International Journal of Database Theory and Application
Vol.10, No.9 (2017), pp.39-46
http://dx.doi.org/10.14257/ijdta.2017.10.9.04

Security Alarm Systems: Modeling and Analysis of SIA Protocol

Shankar Raman Ravindran

Department of Cyber Security Systems and Networks,
Amrita University, Amritapuri, India
shankaraman.r@gmail.com

Abstract

Modeling network specific parameters and conducting analysis on the alarm system protocols helps in predicting an event loss during communication failures. The analysis and the experiment results were conducted on one of the alarm system protocol, Security Industry Association (SIA) protocol. The contribution includes our approach to model the alarm system parameters, a tool that can help in simulation (PE Simulator) and experiment results on the retransmission mechanism deployed by the protocol. This analysis and the results can help a user to tune the device or network in order to prevent the event loss.

Keywords*: Monitoring Station, Premises Equipment, Alarm Systems*

1. Introduction

Nowadays, *Security Alarm Systems* are widely used in many parts of the world. They can be found in places such as: residents (home automation), banks, supermarkets, buildings (building automation), stadiums and more. When appropriate sensors are attached to the system they can prevent unauthorized person entering a protected premise (for example: museums), prevent theft in banks, monitor precious objects inside the galleries, jewelry shops *etc*. Alarm systems are generally monitored by a group of people who provided services to customers by deploying them in their premises and also controlling them from a place called Control rooms or Monitoring Stations (MS). The equipment deployed in the customer premises are collectively called as Premises Equipment or PE. The MS uses special software applications to receive information from the PEs deployed in the customer premises. They are called as Monitoring Station Receivers or MSR. MS staffs uses the software to effectively control, manage, and receive messages from the PE. We use the terminology Events for all the messages that are transferred from PE to the MS. The events may indicate the output of an alarm system or from its components (sensors, bus etc) or it can also be the status information of the PEs.PEs communicate information to the MSR using various application layer protocol specifically designed for them: Ademco Contact ID[1], KNX[2], and SIA[3]. Our focus will be on the SIA protocol.The first analysis on this protocol was done in 2010 [4]. The research work contributed the experimental results, merits and demerits of this protocol. The analysis was performed on the network capture dumps (PCAP files) that were captured between the device and the monitoring station. PCAP files stores all the information transferred between two or more end points in a network. With the help of the PCAP files the researchers found it is possible to trigger different events by replacing the captured traffic without getting notified to the monitoring station. The research work covered the attack vectors that bypass an alarm event from a sensor installed in the security panel, behavior of alarm systems when they are exposed to network vulnerabilities, and improvements to thwart those attacks.The attacks carried out on

Received (June 18, 2017), Review Result (August 13, 2017), Accepted (September 8, 2017)

Google Nest Thermostat device is described in [5]. The problems stems up due to the complexity of the Nest infrastructure that eventually leads to some unnoticed bugs, which can be exploited. Nest uses signed firmware updates but due to the lack of protection a potential attacker can install malicious scripts in the device. The research work demonstrates a technique which can bypass the authentication process of the firmware upgrade process through a malicious USB drive.Modeling a network infrastructure helps in getting a better picture of the elements in the network. It also helps in better utilization and management of resources. Designing alarm systems is described in [6]. The paper presents a framework that can help in designing alarm systems to reduce false alarms and for effective generation of alarm events. By adding delays in alarm, filtering data are some techniques effectively used to reduce these false alarms. Apart from these techniques there are newer ones such as confirmed alarms that can allow user to specify a condition and alarm is raised only when that condition is met.We will model the Alarm System environment using a framework called as A4WSN[7]. The designing models are represented based on three viewpoints: SAML, NODEML, and ENVML. SAML focuses on modeling the software aspects of the environment, NODEML focuses on describing the components of a hardware such as type of processors, memory, storage etc, and finally ENVML focuses on the modeling the obstacles through which the signals are transmitted.When we talk about protocols it is important to discuss the non-functional (NF) properties and attributes. Reliability is the NF property and response time, delays, retransmission intervals are the NF attributes we will be focusing on this protocol. These factors are important because they contribute to the quality of a protocol. Our work will focus on in-depth analysis of non-functional attributes on the recent version of the protocol SIA-DCS-09[3].Our research work is organized as follows: Section 2 will be describing our SIA-PE Simulator's architecture and functionalities. In Section 3 we will be presenting our experiment results on the NF attributes of the protocol. Section 4 describes our approach to destroy the event logs from the panel.

2. PE Simulator - Architecture Overview

The simulator was developed to help developers to understand SIA protocol's behavior and mainly focused on developing a CSR software. Secondly to enable some level of simulation to understand the event loss and retransmission.There are some CSRs (for SIA) available in the internet but there are no SIA PE Simulator which can generate the events for the CSRs. Using this simulator a developer can test their CSR software without purchasing a alarm system panel.A simple Alarm system can made up of the following : Sensors and Cameras, System Panel and Keypads. Sensors and Cameras are the ones that detect intrusions and act as an input to the System Panel. System Panel acts like a CPU in the personal computers. They manage all the sensors, cameras and other devices installed in the system. The sensors, cameras and other devices are connected using the bus and zones. Keypads, allows a user to interface the functionalities provided by the System panel.

It helps a user to configure sensors, set time, arm/disarm sensors, set IP address, MS parameters etc.We have designed our PE simulator architecture based on the simple alarm system described above. The Sensors and Cameras are designed using the OpenHAB framework. Optionally, users are also allowed to write their program to interface the simulator. We mentioned above that the sensor uses bus and zones to communicate the information to the system panel. In our architecture MQTT connections acts as bus/zone and PE Simulator as a part of the system panel. Figure 1 is the UML Deployment diagram for the SIA PE Simulator. The panel specific parameters are modeled in the A4WSN[5] framework. A4WSN uses XML files to store the modeled details. A parser program is used to extract the simulation parameters and will be saved as another XML file. This file will be fed as an input to the SIA-IP PE Simulator software. The simulator uses those parameters for the simulation. The sensors and environment related details will be designed separately using the OpenHAB framework. Optionally it can also be designed using the A4WSN framework and this option is left to the end user to decide.The Simulator uses a MQTT subscriber program to receive the alerts from the sensors configured in the OpenHAB framework. The simulation will be carried out based on the panel specific parameters defined in the modeling phase. The hardware related parameters will be modeled in NODEML, alarm conditions and software specific details and constraints will be modeled in SAML, and finally the obstacles such as the materials we will find in the environment are modeled in the ENVML.

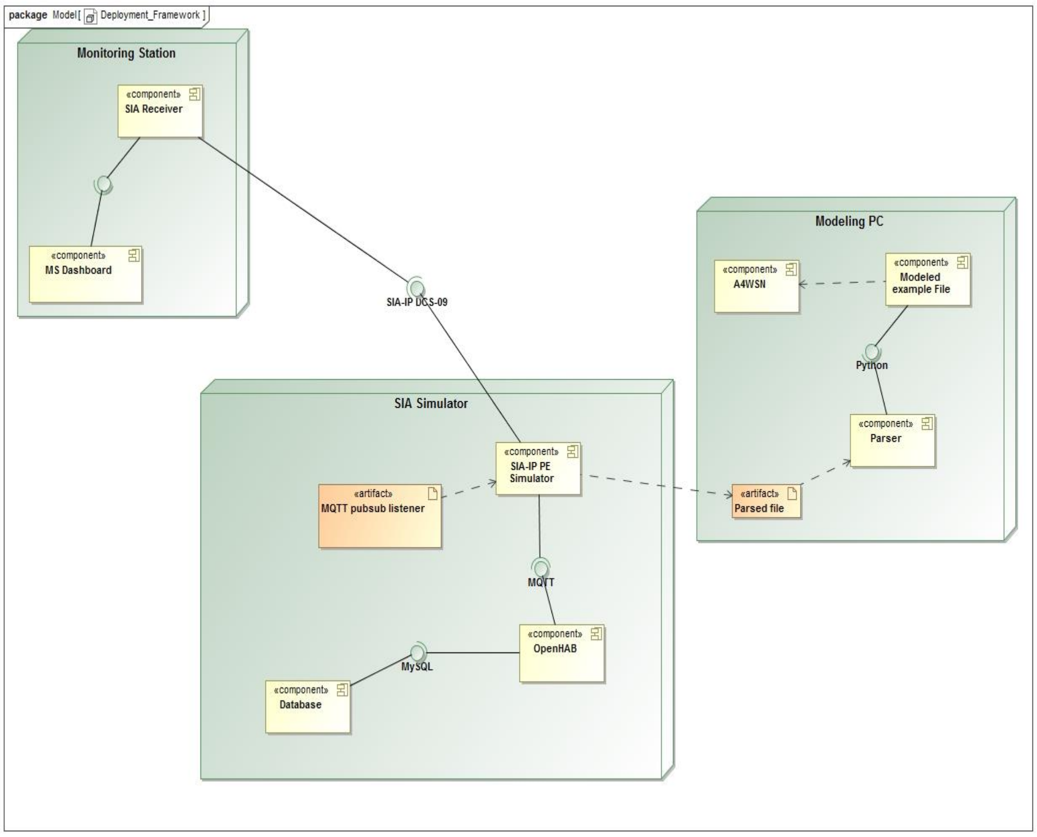


Figure 1. SIA PE Simulator

The simulation software uses the information modeled in the A4WSN framework and performs the simulation. A list of events can be configured separately to enable simulation of those particular events. A SIA-IP Receiver or CSR is another software program which listens to SIA events transmitted by the simulation software. The CSR software program parses the event details, performs the necessary event code mapping and publishes the result to a Dashboard. The flow of events is described in the following UML Sequence diagram.

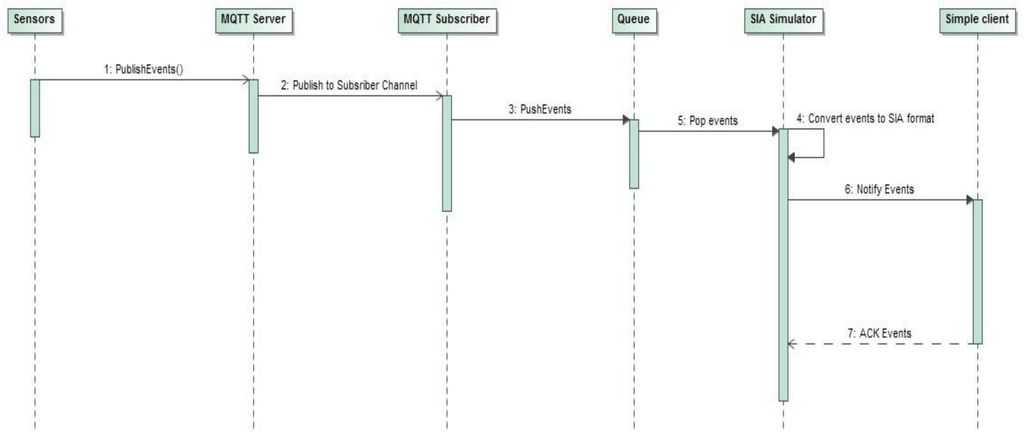


Figure 2. Flow of Events

3. Experiment on Event Retransmission

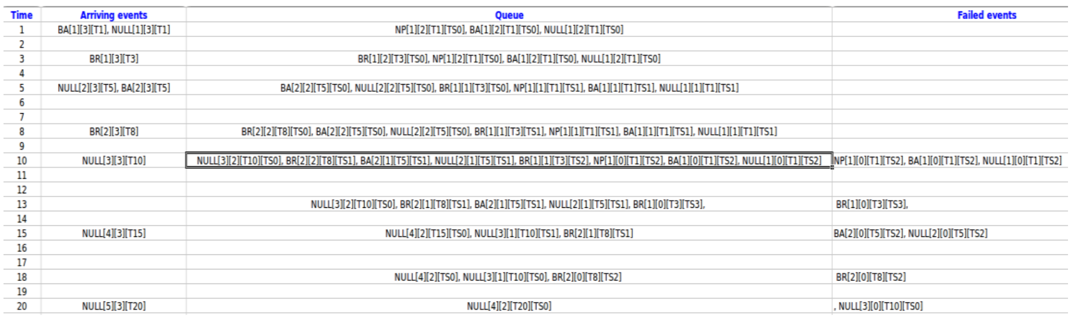
In this section we will discuss our experiment analysis on the re-transmission mechanism adopted by the protocol. We will also discuss the steps that lead us to erase the event logs by leveraging the results obtained from the experiments we conducted on re-transmission. An event re-transmission may occur due to various reasons: faults, errors in network devices, problems in acknowledging the event, power outages, device configuration changes, and due to security attacks such as DoS *etc*.We conducted couple of experiments to understand the queuing mechanism deployed by the device to handle network loss and also to understand the re-transmission mechanism. With the experiment results we were able to destroy some part of the evidences created in the device. We meant the Event Log.

3.1 . Experiment

Upon transmitting an event Premises equipment waits for a response from the central station receiver. SIA recommends the retry timeout period to be 20 seconds and the number of message attempts is 3. But the retry timeout can also be configured in the range of 5-60 seconds. The experiment was conducted on a system having a retry timeout of 7 seconds and maximum retries of 15 times. The Monitoring station menu in the panel was configured with the following parameters:

- No of retries : 2 (Excluding the 1st attempt)
- Time interval of retries : 5 second
- Polling frequency : 1 packet every 5 seconds
- Alarm restore : Immediately after 3 second from the start of an alarm.

As per SIA, the Validity of an event = 60 seconds x 3 attempts = 180 seconds. That is, if the network failure is not resolved within 180 seconds (3 minutes) the events which were queued will be completely lost.The network cable between the premises equipment and the central station receiver was tampered. With network connection unplugged, a burglary alarm was generated from a volumetric sensor present in the panel. We discovered the 1st re-transmission was made at 7 seconds and the 2nd in 14 seconds by sniffing the traffic at the PE side. After 14 seconds the connection was brought up and we never got the burglary alarm. Figure 3 is the screen-shot displaying the state of various events at different time period.



Time	Arriving events	Queue	Failed events
1	BA[1][3][T1], NULL[1][3][T1]	NP[1][2][T1][TS0], BA[1][2][T1][TS0], NULL[1][2][T1][TS0]	
2			
3	BR[1][3][T3]	BR[1][2][T3][TS0], NP[1][2][T1][TS0], BA[1][2][T1][TS0], NULL[1][2][T1][TS0]	
4			
5	NULL[2][3][T5], BA[2][3][T5]	BA[2][2][T5][TS0], NULL[2][2][T5][TS0], BR[1][1][T3][TS0], NP[1][1][T1][TS1], BA[1][1][T1]TS1], NULL[1][1][T1][TS1]	
6			
7			
8	BR[2][3][T8]	BR[2][2][T8][TS0], BA[2][2][T5][TS0], NULL[2][2][T5][TS0], BR[1][1][T3][TS1], NP[1][1][T1][TS1], BA[1][1][T1][TS1], NULL[1][1][T1][TS1]	
9			
10	NULL[3][3][T10]	NULL[3][2][T10][TS0], BR[2][2][T8][TS1], BA[2][1][T5][TS1], NULL[2][1][T5][TS1], BR[1][1][T3][TS2], NP[1][0][T1][TS2], BA[1][0][T1][TS2], NULL[1][0][T1][TS2]	NP[1][0][T1][TS2], BA[1][0][T1][TS2], NULL[1][0][T1][TS2]
11			
12			
13		NULL[3][2][T10][TS0], BR[2][1][T8][TS1], BA[2][1][T5][TS1], NULL[2][1][T5][TS1], BR[1][0][T3][TS3],	BR[1][0][T3][TS3],
14			
15	NULL[4][3][T15]	NULL[4][2][T15][TS0], NULL[3][1][T10][TS1], BR[2][1][T8][TS1]	BA[2][0][T5][TS2], NULL[2][0][T5][TS2]
16			
17			
18		NULL[4][2][TS0], NULL[3][1][T10][TS0], BR[2][0][T8][TS2]	BR[2][0][T8][TS2]
19			
20	NULL[5][3][T20]	NULL[4][2][T20][TS0]	, NULL[3][0][T10][TS0]

Figure 3. Simulation Table Results

3.2. Table Explanation

Each cell has event codes that are sent by the PE to MS.

- BA Burglary Alarm • BR Burglary Alarm Restore
- NULL Polling message.

Column shows the state of the events at different time (t). Details encoded in the event code index has meanings with respect to the column.

1. Column 1 : Indicates the time
2. Column 2 : Contains information about the events generated at time (t) from column 1.
   Event Code [*event number*][*remaining retries*][*time of the event*]
3. Column 3 : Contents of the devices network queue. The events are queued for retransmission.
   Event Code [*event number*][*remaining retries*][*time of the event*][*timestamp*]
4. Column 4 : Events that are failed to transmit. The events shown here will not be resent. In other words, the generated events are lost.
   Event Code [*event number*][*remaining retries*][*time of the event*][*timestamp*]

3.1. Example

The *remaining retries* index of an event will be decremented by 1 when it is added to the queue. The device in our lab follows a LIFO (Last in First Out) implementation. Every time the device tries to resend an event it attaches a different timestamp based on the *interval of retries* configured. The following points will explain the details of the Fig.1.3 by taking network failure in consideration.

1. At t=1, a burglary alarm (BA) and a polling message (NULL) was generated. Since there was anetwork failure, the events were queued for retransmission.
2. At t=3, the burglary alarm restore (BR) was generated and added on top of the queue (LIFO)
3. At t=5, the events generated at t=1 will loose their first timestamp. They will be added back to thequeue with a new timestamp (initial timestamp + *time interval of retries*) and the *remaining retries* count will be decremented. Also, there are new events: a burglary alarm and polling message being generated and added to the queue.
4. At t=8, the burglary alarm restore was genereated for the burglary alarm generated at t=5. At thissame point of time, the BR generated at t=3 is added back to the queue with a new timestamp and it's *remaining retries* count decremented by 1.
5. At t=10, the events which are generated at t=1 were retransmitted and failed. Since the*remaining retries* count is now 0, these events will not be transferred again. In other words these events are lost. The failed events are added to *Column 4*. *Column 4* is just to track the failed events. Device doesn't store them. They will be erased from the queue.

This fact must be taken into account when a user decides to choose the SIA protocol.. We also noted that some device manufacturers customize the MS parameters such as *no of MS retries, and interval between the retries* different from SIA's recommendation. It is important to keep these parameters in mind and tune the device parameters accordingly.

4. Experiment on Destroying the Event Logs

In order to prevent loss of events device manufacturers use Event log, Cloud, GPRS as back up mechanisms. Unfortunately the event log entry sizes are restricted. For example, our devices support up to 250 entries. Event log works in a circular fashion. That is, when the size of an event log has reached the limit, the newly arriving entries will overwrite the first entry.Fortunately the information about the event was logged inside the Event Log manager of the device. Our next experiment was to further take this experiment to wipe out all the existing entries in the event log. By this way we can override the Burglary alarm event with a random event in the Event log. We found the device event log size to be 250 entries and as mentioned above they work in a circular fashioni.e after 250 entries the 251st entry will over write the 1st entry in the Event log.Apart from logging events related to alarms, the device also logs other miscellaneous events. In most of the devices, when there are system troubles like communication issues, power outages, battery outages they will be immediately logged as an entry inside the Event Log. Our attempt is to overwrite the Burglary alarm event with these entries that can be generated using one of the above problems.Practically it is a difficult task to physically unplug/tamper the network connection deployed in the premises. They might have tight security measures employed in the premises. For example, we can find sensors like Volumetric, Security cameras, and staffs being deployed in the premises for protection.The next possible option would be to do a DoS attack on the premises equipment and the network to prevent PE to transmit any events to the CSR. In other words, we flood the victim's network with maliciously crafted packets. The type of DoS attacks we used are mentioned in the background section.As we all know, Security monitoring Systems are not really powerful like the computers/laptop we use in our daily life. Sometimes when there is a lack of protection to thwart DoS attacks, a device may crash. But the device we used had SYN cookies enabled which would effectively thwart DoS/DDoS attacks. But it cannot prevent the flooding of incoming packets unless very strict measures are deployed.In spite of the protection (SYN cookies), we managed to flood the victims network with crafted packets. In effect, we observed that every time when the device tried to transmit the event it eventually failed due to the flooding of packets. This status was logged as a Network communication trouble inside the Event Log. We noted that this message was created every 7 seconds inside the Event Log. The device already had 77 entries which were not cleared. The burglary alarm was logged in 78th and the alarm restore in 79th position.

- Total no of event log entries = 250
- No. of existing entries = 79
- No. of entries to be overwritten = 250 79 = 171
- Time interval to log a Network communication trouble = 7 seconds

Given this information we have to run our scripts and tools for 25 seconds (171/7) to exhaust the maximum space (250 entries) provided by the Event Log. We flooded the network by using all the techniques mentioned in the background section: *SYN-Flood, ICMP flood and also Ping of Death*. SYN-Flood was carried out by using libnet-library (in C) and the others were done using *hping3* utility.Following are the syntax and parameters for the *hping3* utility to perform *ICMP flood and Ping of Death* DoS attacks.

ICMP flooding :

```
$          hping3 −C 500000 −−icmp −i −u1000            10.0.40.13
```

Ping of Death :

```
$                        hping3 −C 500000 −−icmp −i −u1000 −d 65536243210.0.40.13
```

-d flag specifies the packet length and -u flag specifies the rate at which the packets will be sent.

5. Conclusion & Future Works

Event Loss problem can be solved by deploying extra back up mechanism such as GPRS, Increased Event Log sizes, Cloud, and a way to store event logs in the customer premises. This thesis is focussed on 3 main topics

- A SIA-PE simulator as a 3rd party plugin for A4WSN.
- Detailed analysis on Event retransmission
- Our experiments on erasing the logged events in the panel

We are yet to develop some part of the architecture (Figure 1) in order to do a complete simulation of alarm events. The development will be made in the following parts:

- Merging OpenHAB framework with SIA PE Simulator • Implementation of Queue to demonstrate loss of events
- Representation of output from the simulation.

References

[1] SIA DC-0-1999.09 Ademco Group **(1999)**.
[2] KNX Association Home and Building Management Systems.
[3] ANSI/SIA DC-09-2013: Internet Protocol Event Reporting. Security Industry Association (SIA).
[4] Modern age burglary, Kevin de Kok, Jeroen Klaver, University of Amsterdam **(2010)**
[5] G. Hernandez, O. Arias, D. Buentello and Y. Jin, Security in Silicon Laboratory, University of Central Florida. Smart Nest Thermostat: A Smart Spy in your Home.
[6] I. Izadi, S. L.Shah, D. S. Shook, S. R. Kondaveeti and T. Chen, University of Alberta. A Framework for Optimal Design of Alarm Systems. Proceedings of the 7th IFAC Symposium on Fault Detection and Safety of Technical Processes, **(2009)**.
[7] K. Doddapaneni, E. Ever, O. Gemikonakli, I. Malavolta, L. Mostrada and H. Muccini, “A Model-Driven Engineering Framework for Architecting and Analyzing Wireless Sensor Networks”, Third InternationalWorkshop on Software Engineering for Sensor Network Applications (SESENA), **(2012)**.