Report for the CPS Mid-Course Project

Andrea Auletta
andrea.auletta@studenti.unipd.it

Niccolò Zenaro
niccolo.zenaro@studenti.unipd.it

December 28, 2024

Contents

1 Introduction and objectives

In this report we will describe the work done for the mid-course project of the course Cyber-Physical Systems and IoT security. The paper to which we refer is [1] **"Fingerprinting Electronic Control Units for Vehicle Intrusion Detection"** where is explained the design of an algorithm: the CIDS (Clock-based Intrusion Detection System), that is able to detect different kinds of attack. In this paper is shown the functioning of three kinds of attack: the "Fabbrication attack", the "Suspension attack" and the "Masquerade attack". What we tried to do was to implement the three attacks and the CIDS algorithm in a simulation environemnt using **ICSim**, instead of physically as shown in the paper.

2 System setup

We worked on a virtual machine with **LinuxMint22** as operating system. The simulator used is **ICSim**. All the code is written in **Python** and can be found at the following link: .We used the **can library** to create messages and to simulate the communication of the attackers with the ECUs.

3 Experiments

For each attack we are assuming that the attacker is already in the system and is able to send messages to several ECUs. To emulate a periodic ECU so that you can at least see something in the simulator we created the script *fakeECU.py* where are sent messages every 2 seconds to open and close a certain number of doors of the car.

3.1 Fabbrication attack

In the Fabbrication attack we are trying to inject messages into the CAN bus to make the system behave in a different way. Essentially, we created a message with the ID of the doors in a way that in the middle of those two seconds gap between legitimate messages we sent other commands to open and close the doors and causing it to function differently from normal behavior.

3.2 Suspension attack

In the Suspension attack we are trying to stop the receiving of messages sent by a legitimate ECU. We made a DoS like attack: we sent a big quantity of messages with the ID of the left arrow into the CAN bus in order to lose the messages sent by it. Here we put a very small time sleep, precisely because the aim is to clog the network and not receive packets from legitimate ECUs. The result was that if we were trying to turn on the left arrow, it was not working.

3.3 Masquerade attack

In the Masquerade attack we are trying to send messages with the ID of the legitimate ECUs. The idea here is that we have to shield the fact that an ECU is compromised. After running the weak ecu script, the attacker will listen the CAN bus for 20 seconds, in this way it can calculate the period of the messages sent by the weak ECU and then it will send messages with the same ID and the same period. We decided to send a message to open different doors with respect to the ones opened by the weak ECU, and this can be seen at simulation time.

3.4 CIDS algorithm

CIDS is a new type of Intrusion Detection System (IDS) that is Clock-based. The algotithm measures and exploits the interval of periodic messages for fingerprinting ECUs. These fingerprints are then used for deriving a clock-behavioural scheme obtained by the Recursive Least Squares (RLS) algorithm. RLS is applied to timestamps and their offsets to derive covariance and skewness. Based on this scheme, CIDS uses the cumulative sum to detect shifts in the clock skew, in fact the intrusion is detected monitoring this parameter. We worked direcly on the dumps of the CAN bus messages obtained through the command *candump -l vcan0*. In the file *cids.py* we implemented the algorithm, this is composed by several parts:

1. Data preprocessing: we've taken the timestamps of the messages and we've inserted them in a vocabulary in which you can access with the ID of the ECU;

2. Offset calculation: the offset is given by the difference of the timestamps and the corresponding expected time (calculated with the period of the messages);

3. Recursive square least algorithm: we've implemented the RLS algorithm to calculate the covariance and the skew;

4. CUSUM algorithm: we've implemented the CUSUM algorithm to detect the intrusion.

The result at the end is the list of ID of the ECUs with an indication which says if the ECU is compromised.

4 Results and Discussion

As said before we've implemented the three attacks and the CIDS algorithm in a simulation environment. We've taken different dumps of the CAN bus messages and we've analyzed them with the CIDS algorithm. The main problem we've encountered is that the CIDS algorithm detected intrusion also on the ID which are not target of the attack, probably this is due to the fact that in the simulation, by sendind a certain quantity of messages the timestamps are not so precise and will be shifted of a certain quantity of time, in this way the periodicity of the messages is not respected and the CIDS algorithm will detect an intrusion. We tried also to change and tune the parameters like the threshold and the k value for to try to give more or less tolerance on the offsets but didn't work as expected. We've got 4 dump files of the CAN bus messages:

- *noAttackDump.log*: this is the dump of the CAN bus messages without any attack. Here we found out that with reference to the other dump file with the attack, the number of the ID with no intrusion detected is quite lower (20/36 have intrusion detected);

- *dump_fabr.log*: this is the dump of the CAN bus messages with the Fabbrication attack. Here 33/34 ID are detected as compromised. Also the ID of interest is detected as compromised (419 = 0x19B);

- *dump_susp.log*: this is the dump of the CAN bus messages with the Suspension attack. 25/36 ID are detected as compromised. Also the ID of interest is detected as compromised (392 = 0x188);

- *dump_masq.log*: this is the dump of the CAN bus messages with the Masquerade attack. 32/37 ID are detected as compromised. Also the ID of interest is detected as compromised (419 = 0x19B).

The results are not like the expected ones, but we think that the problem is due to the fact that the we had problem with the precision of the timestamps and the periodicity of the messages. A factor to keep in mind is that the dump without attacks had a lower number of ID detected as compromised, and this can be used as a reference to understand if the system is under attack or not.

References

[1] Kyong-Tak Cho and Kang G. Shin. "Fingerprinting Electronic Control Units for Vehicle Intrusion Detection". In: *Proceedings of the 25th USENIX Conference on Security Symposium* (2016), pp. 911–927.