Obstacle detection and classification : Attacks and Risk Mitigation

Abstract

In this article, we focus on the security analysis of our project : detection and classification of objects using mmwave radar.
Our project has not developed any security mechanism which, itself, constitutes a major vulnerability. Here we describe different attacks and we suggest mitigation policies to improve the security level.

Sami BEYAH & Yosra ZEYRI NEMRI
MSIoT

Introduction

We worked for ACTIA on **using radar technology for obstacle detection and classification.** The aim of the project was to give driving assistance to aerial bucket's operators in manufacturing sites, by displaying an accurate position of the surrounding objects in the near area (up to 30 meters). The code implemented to detect obstacles has been written in C-language and embedded in the radar. We had to test the radar in different conditions, optimize the detection parameters and review the code to include our values for a proper detection. The second purpose was to classify in real-time these obstacles (humans,animals, other vehicles, walls etc.). We added a camera right above the radar and used the Yolov5 python program for object classification. The purpose here is to combine both obstacles classification and position measurement.

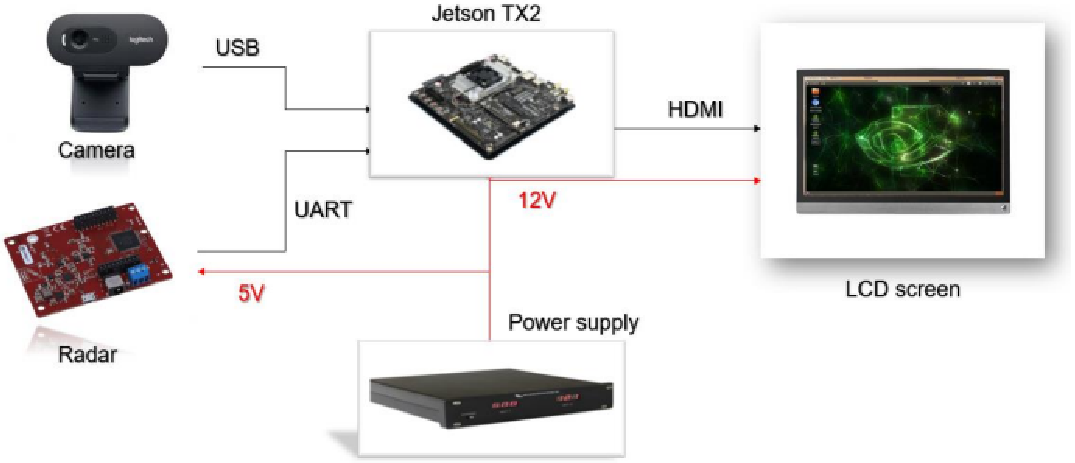

Figure 1 : Overview of our embedded system architecture

Our system can be used by operators to make navigation decisions so the information provided by the sensors (camera and radar) should be reliable. Attackers can use a security vulnerability to feed erroneous sensor data with the intention of disrupting or taking control of the system. In this paper, we analyze all vulnerabilities in our system that an attacker might exploit and we propose countermeasures.

The threats can be classified through a three-tier hierarchical system as shown in Figure 2 below. We would analyze possible security threats on all three layers.

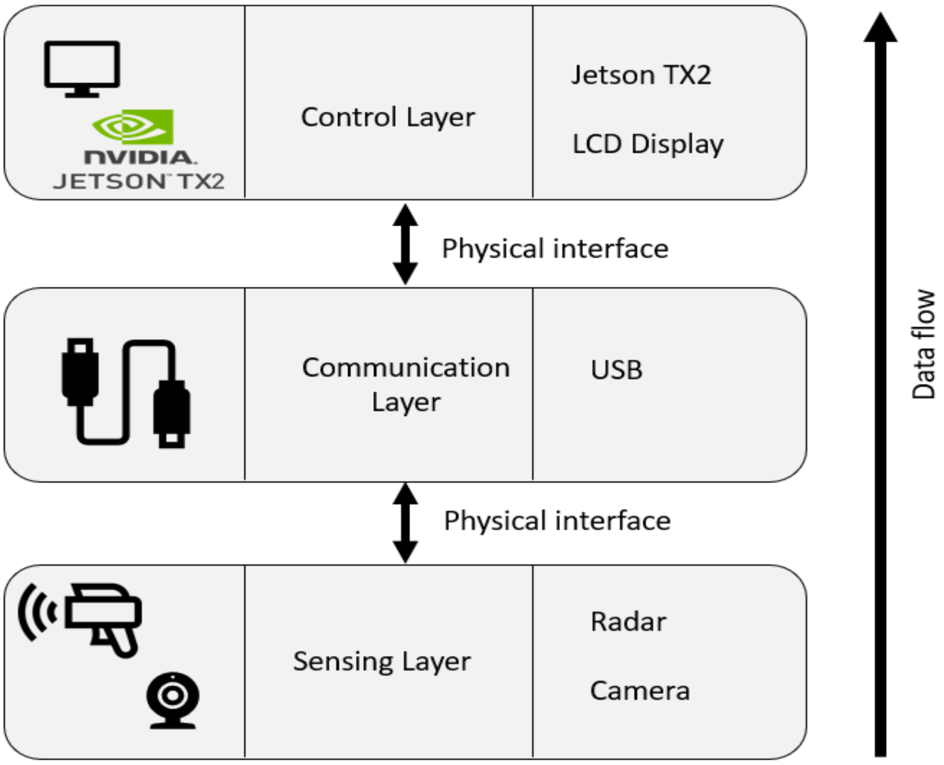

Figure 2 : Attack surface

I- Evaluation of the security risks : Vulnerabilities

Our embedded system is a standalone system i.e an embedded system that is not part of any network. Our device is not connected to the internet because it does not include network communication protocols such as TCP or UDP. Likewise, it does not have wireless communication modules. Therefore, the system cannot be remotely attacked and can only encounter “physical attacks". In this paper, we will focus on those attacks.

In the litterature, we have identified different security risks in our systems and some countermeasures. Will will only focus on attacks that might alter the integrity or the availability because the confidentiality is non-critical. This is a non-exhaustive list :

Risk name	Countermeasure
Jamming radar	Noise detection and rejection; Multiple sensors for redundancy check
Spoofing radar	Noise detection and rejection; Multiple sensors for redundancy check

Relay attack	Noise detection and rejection; Multiple sensors for redundancy check
Blinding cameras	Multiple cameras; Filter to remove harmful light
Code injection	Device authentication; Isolation
malicious USB cable	Don’t just plug in any USB cable, always use trusted hardware
Malicious hardware replacement	Encryption, strong packaging

A. Sensing Layer attacks

Our system is equipped with 2 sensors. A radar sensor to detect objects distance, angle, and velocity. And, a Logitech camera for object classification using Yolov5. Both sensors display detected objects in real-time. An attacker must attack both sensors, which increases the cost and complexity of the attack. We will study both possible physical attacks on radar and on cameras.

1. Radar security vulnerabilities

First, we will focus on security vulnerabilities due to the radar. Our studies showed that millimeter wave radar can suffer from :

- Jamming attacks can make detected objects disappear.
- Spoofing attacks can alter the object position.

To carry out these attacks, the attacker must be able to identify (by using a signal analyser) the frequency band, modulation scheme and waveform pattern of the radar. Then the attacker can jam and spoof the radar sensor with electromagnetic waves in the same frequency band by a signal generator. **As far as we know, there is no commercial anti-jamming / spoofing millimeter wave radar available in the market.**

However, attacking a radar is very difficult, there is a huge knowledge threshold to be reached by the attacker. He must understand the system model, working principle, relevant physics, and skills to build or operate hardware equipment. Moreover, to carry an attack against a millimeter wave radar, the attacker must use very expensive hardware equipment. The best signal analyzers and generators can only reach 40-50 GHz, frequency multipliers and mixers must be used. An attack against a radar can cost up to 60 thousands dollars in hardware equipment (our estimation based on the price of these equipment).

1.1. Jamming attacks

We programmed our TI AWR1843 radar to work on the frequency band 76-77 GHz. Using a very performant (and expensive) spectrum analyser, an attacker with access to the radar can identify the frequency band, the bandwidth (ramp height) and chirp sequence.

After knowing the waveform parameters, an attacker can jam the radar within the same frequency band. The attacker can send a much more powerful signal compared to the real reflected signal arriving at the radar.

This attack will lead the radar system to failure. The radar will not be able to detect objects anymore as long as the attacker is sending electromagnetic waves. This attack focuses on compromising availability.

This attack can be easily detected if countermeasures are implemented in the code.

1.2. Spoofing Attacks

Using a radar of the same type or an SDR per example, an attacker can falsify the radar measurements if he can identify the waveform parameters and duplicate them. From then on, an attacker can display false information of distance or velocity on the operator display. Spoofing makes it very difficult for the sensor system to recognize that it is under attack, as it provides the radar with seemingly legitimate but actually false data.
A simple spoofing attack is : an attacker replays a counterfeit signal with additional physical delay (τ) to create an illusion that the object is further away than the actual distance.
This attack has been perpetrated several times on radar using FMCW modulation. The attacker can also falsify the velocity of detected objects or show phantom objects.

1.3. Relay attacks

Relay attack is very similar to a spoofing attack. The difference is that the attacker will not try to falsify the data like in a spoofing attack. The attacker will store the radar emitted signal, create a digital duplicate, and then continually retransmit the duplicate so that the radar will consider it to be a legitimate signal. Thus, the radar will not be able to detect objects because he will receive the same duplicate of the signal over and over. This will lead the operator to think there are no new detected objects and can lead to collision.

1.4. Countermeasures

In the litterature, we found out that **the best countermeasure for these attacks against radar technology is incorporating data fusion and attack detection.** In this paper [1], the authors propose using multiple sensors for redundancy check. Having another radar makes it easy to discard invalid inputs and keep our system working even if one radar is attacked. The cost to implement this is the price of an additional sensor (around 350 euros).

For Jamming attacks, they can easily be recognized because there are very few sources of millimeter wave radio noise in the working environment, especially with high power that can make measurements impossible. We can add noise rejection to detect and filter jamming attacks. The only cost for this kind of approach is the implementation of the algorithm, which is relatively low.

In case of spoofing and relay attacks, we found that we can use hybrid filters that use the modified Kalman filter and Chi-squared detector to detect false data that is injected at random points and minimize the impact this data have on radar sensor inputs.

2. Camera security Vulnerability

We use a Logitech camera to display object classification on the operator. To do so, we run the classification algorithm on the Jetson TX2. Depending on the detected object classification, the operator of the aerial bucket will take navigation decisions. It is crucial that the data displayed is reliable.

2.1. Blinding attack

Cameras are passive light sensors. From our daily experience, they can be blinded or fooled in many ways. Blinding attacks are the easiest and most common attacks on cameras.

The attacker disables the functionality of the system camera sensor. A strong laser beam focused at the camera leads to higher tonal values, and the attacker exploits this phenomenon to conceal the camera feed, causing complete blindness to the system sensory inputs. LEDs can also be used to generate bright light against cameras. Photoelectric sensors in the camera are very sensitive to the intensity of light. LED or laser attack can also lead to irreversible damage to the CMOS/CCD chip inside the radar.

This attack will lead to system failure because the system will not be able to display or classify objects. Our Logitech camera does not provide noise reduction or protection, and thus can be blinded or permanently damaged by strong light, which will further lead to failure of camera based-functionalities (object classification).

2.2. Countermeasures

Countermeasures exist to protect cameras from being tampered with. We looked for a trade-off between protecting the camera from tampering, sensitivity, image quality, camera size and price. Most of the countermeasures we found required the camera to be modified. This will significantly increase the cost.

The best trade-off we found is using multiple cameras that perceive the same image, the attacker has to put more effort into the attack to blind all cameras at the same time.

Another take is using a filter to remove harmful light sources. The filters can vary depending on the use cases, some can only filter one type of light while others are made from materials capable of filtering multiple sources and change its color and opacity depending on the input.

B. Communication Layer attacks

In this section, we will look at threats on the communication layer of our system: between the sensors and the control/application unit and between the control/application unit and LCD screen. All communications are wired: USB and HDMI.

3. Communication Layer security vulnerabilities

3.1. Malicious USB attacks

In the wiring of our system architecture, we need to plug in a radar, Logitech camera, and LCD screen and connect them to a Jetson via a USB/HDMI cable.

However, this comes with a risk that our system will be damaged with Malicious USB cable which looks like a perfectly normal USB cable. We would use it to plug into our computer and connect our devices to the computer, except it has some additional electronics inside. It could, in fact, tell our operating system that it is a HID, or a human interface device. This is the categorization for keyboards and a mouse, so when we plug in this malicious USB cable, it’s able to start typing anything that it

would like into our system. So, we might plug in the cable, it would start up a command prompt, it would type in some commands to download some malware from a third-party site, and then our system is infected.
As it concerns HDMI cable, The worst a malicious hdmi cable could do is to send the video to someone else. It can't alter the integrity of the signal or be used to carry an attack on the Jetson TX2.

3.2. Countermeasures

This sort of malicious activity is almost impossible to detect through conventional means, as virus scans done with machines infected via the USB exploit will turn up nothing. The researchers have found that the only way to effectively figure out whether a device is infected is to take it apart and reverse engineer it.

But, to avoid USB-based attacks, we shouldn't simply plug-in any USB cable that we happen to find. We need to have some trusted knowledge of where this cable came from, and trust that it came from a reliable source. And then to ensure our data security we can encrypt sensitive data (To protect our data is to encrypt it). In order to access encrypted data, a user has to enter a password or key file. Even if attackers manage to steal your information with a USB cable, they won't be able to use it if it's encrypted.

C. Control/Application Layer attacks

The control Layer is the most critical part of our system. It is composed of the Jetson TX2 and a LCD display. We used a Jetson TX2 by NVIDIA to configure then recover and process in real-time the data sent by the sensors. We used Jetson TX2 because it offers high computation power and high memory storage. And, a LCD screen to display the object classification and coordinates to the operator.
We did not incorporate any security measures inside the Jetson TX2. We will focus here on attacks against the control Layer.

3.1. Malicious Hardware component replacement

Because our system does not incorporate any security measure, a very trivial attack is to simply replace a hardware component from our system with a malicious one. The Jetson TX2 or the LCD screen can be replaced by a malicious component. The system will not show an alert or stop functioning. Our Linux based Jetson TX2 is only protected by a password. To defend against attacks, we found in the litterature that approaches such as encryption, secure flashing, anomaly and intrusion-detection have been developed that detect and prevent physical attacks.

3.2. Code Injection

Another dangerous attack against our system is code injection. An attacker can simply dump our code and add malicious code to alter the functioning of our control Layer. This is possible on the radar or camera (sensing layer), and on the Jetson TX2 (control Layer).

3.3. Countermeasures

An obvious solution is to use a **strong packaging** to protect the hardware components. Another solution in the literature is to use encryption schemes. For example, we propose using ECDH to pair sensors before starting to retrieve data. The Jetson TX2 and radar AWR1843, will both compute a pair of keys. They will share their public key and compute a shared secret using ECDH algorithm.

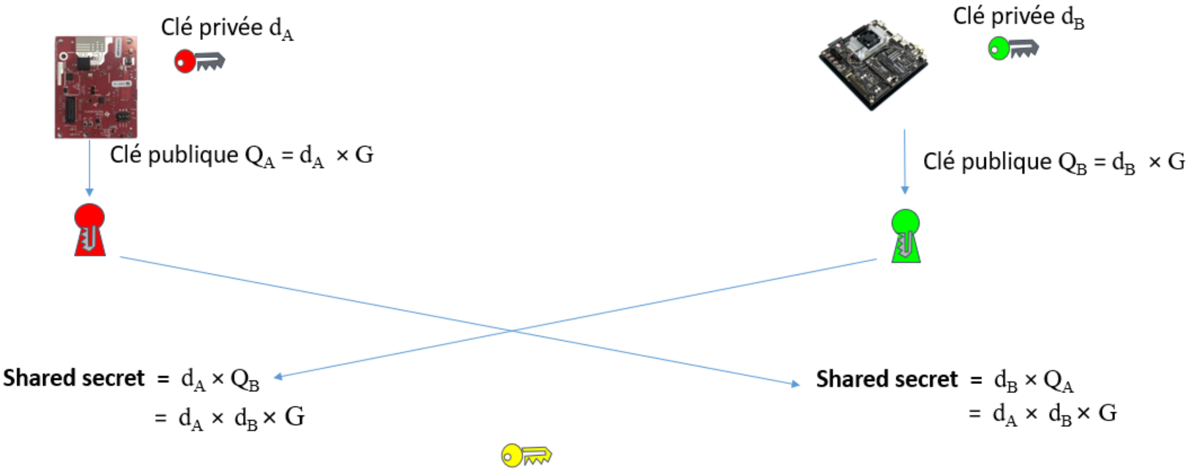
Clé privée d_A
Clé privée d_B
Clé publique $Q_A = d_A \times G$
Clé publique $Q_B = d_B \times G$
Shared secret = $d_A \times Q_B$
= $d_A \times d_B \times G$
Shared secret = $d_B \times Q_A$
= $d_A \times d_B \times G$

Figure 2 : ECDH

Then, we propose using HMAC-SHA256 to verify that the data flow is reliable. Every message will be signed using HMAC-SHA256 using the shared secret. The signature will be sent along the message on plaintext.

On the control layer and on the sensor, both parties will verify the integrity of the data. Every message will be verified by both parties to make sure that there is no attack. The verification will be done on a dedicated thread in parallel to not decrease the system performance.

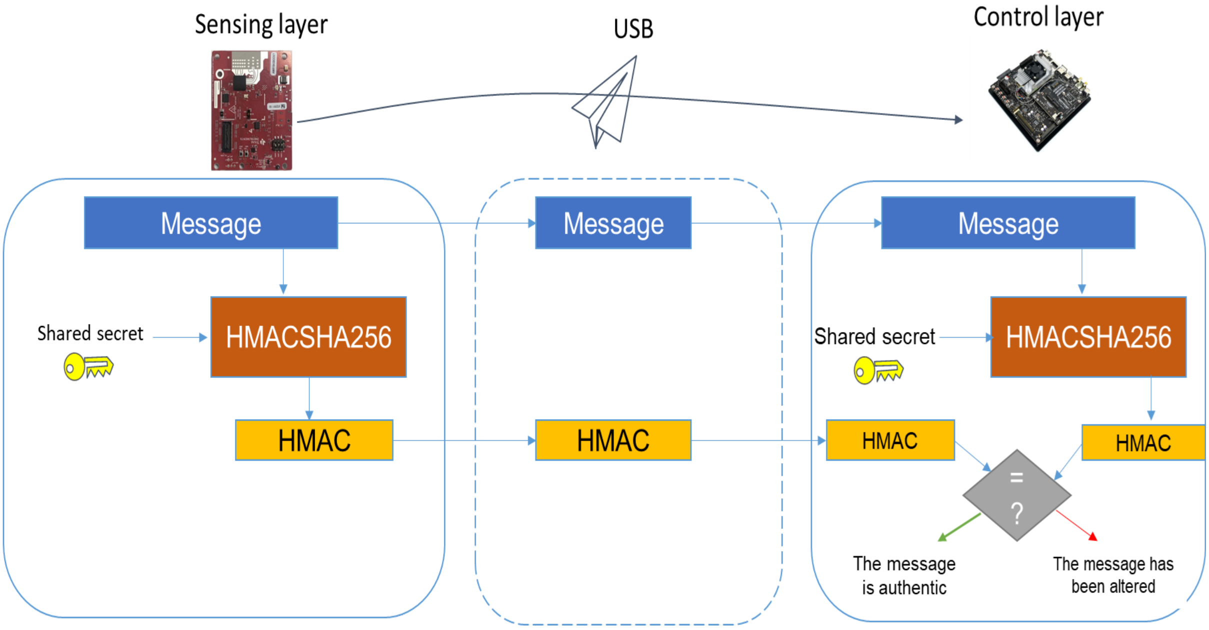

Figure 3 : ECDH + HMACSHA256

However, this approach can decrease the system performance by increasing latency. M. Cayre helped us find another encryption method that can be less time consuming. We can encrypt the data flow using the same key on both sides and using a PRNG.

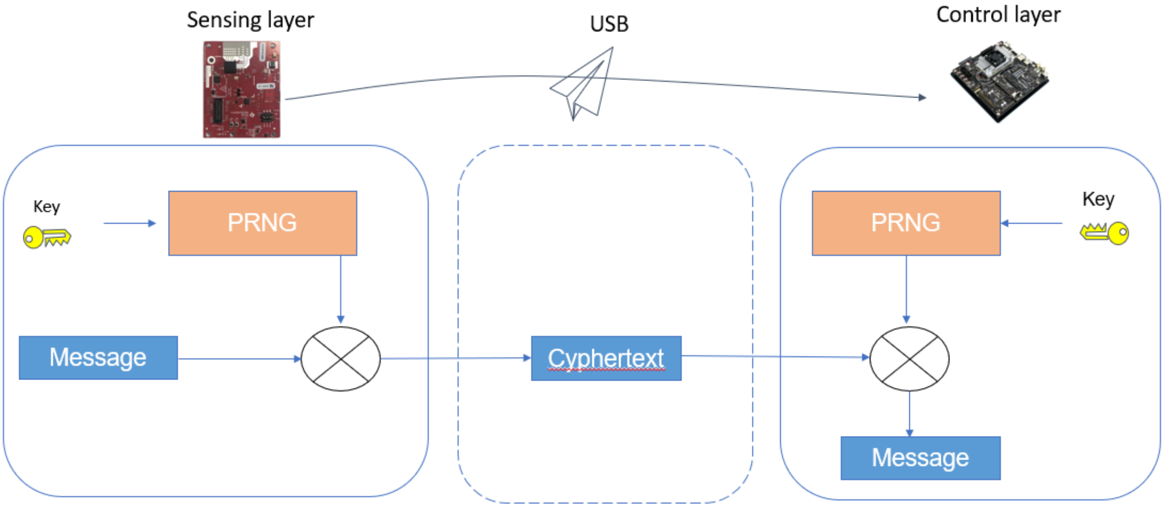

Figure 4 : Flow encryption

This method can ensure data integrity and is not time consuming.

The best countermeasure against code injection is to sign the code using a cryptographic primitive. We propose using ECDSA. We will sign the entire code with our private key and store the public key in a secure storage space in the ROM. When the code is launched, the Bootloader will verify the code using the signature and our public key, if the code has been altered, the code will not be launched.

Conclusion

In this paper, we focused on threats on the embedded system we developed for the innovation project module. The system does not include any security which makes it highly vulnerable. We proposed countermeasures to protect the integrity and availability of our system.

References

1. Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-driving Vehicle : https://link.springer.com/content/pdf/10.1007%2F978-3-030-79108-7_2.pdf
2. GhostImage: Remote Perception Attacks against Camera-based Image Classification Systems : https://www.usenix.org/system/files/raid20-man.pdf
3. IT Security In Radar Sensor Systems - A Methodical Approach : https://www.researchgate.net/publication/352939146_IT_Security_In_Radar_Sensor_Systems_-_A_Methodical_Approach
4. Spoofing Attacks to Radar Motion Sensors with Portable RF Devices: https://ieeexplore.ieee.org/document/9360393
5. Vulnerability of Radar Protocol and Proposed Mitigation: https://www.researchgate.net/publication/311480424_Vulnerability_of_Radar_Protocol_and_Proposed_Mitigation
6. Spoofing Attacks Against Vehicular FMCW Radar : https://arxiv.org/pdf/2104.13318.pdf
7. Security for Safety: A Path Toward Building Trusted Autonomous Vehicles : http://jin.ece.ufl.edu/papers/ICCAD_18.pdf
8. USB-Watch: a Generalized Hardware-Assisted Insider Threat Detection Framework: https://www.researchgate.net/publication/339632610_USB-Watch_a_Generalized_Hardware-Assisted_Insider_Threat_Detection_Framework