Security expectations versus limitations

In part 2 of this article series, we look at possible regulations for future embedded networks. What can developers of Classical CAN and CAN FD based systems do to minimize the potential impact of future regulation on their design?

If you are following technical news around the globe, you can easily get the idea that it is just a matter of time until we will have fleets of autonomously-moving vehicles of all kinds: driving, flying, swimming, or diving. Some of these will be small, like a drone to explore weather parameters or to optically check the condition of a construction. Some will be very big like a road truck or a freighter ship. Even with the intelligence needed for autonomous operation built-in, these vehicles will always need some information exchange with the outside world. At a minimum they will have a command and status interface, but more likely they will have to share plenty of information about their environment including other vehicles in their vicinity.

This data would be processed to calculate the best route to take and to coordinate the routes of all vehicles that are currently moving in the same area. For security reasons, there will likely have to be a mode to manually take over control of the vehicle by an operator in some service center if something fails and a vehicle needs to be taken out of harm's way. This could be directing a malfunctioning car off the road or finding a suitable emergency landing spot for a failing drone.

Adding more communication interfaces and command levels to such vehicles has one downside: each of these is a potential attack vector for hackers. And for hackers a target becomes more attractive the more devices there are. Just imagine there would be:

- fleets of autonomous cargo freighter ships,
- fleets of autonomous passenger cars,
- fleets of autonomous freight trucks,
- fleets of autonomous delivery drones.

Such systems attract all kinds of hackers, including those that try to extort money with ransomware, terrorists, and the "because I (think I) can" crowd. Once the first vehicle is hacked, all other vehicles using the same security methods are at risk as they share the same vulnerabilities. And before you know it, an attacker could have operating control over an entire fleet of vehicles. We predict that not before long you will see a Hollywood blockbuster movie that picks up on these scenarios. Imagine bad guys in control of a fleet of drones hunting down their victims from the air or in control of a fleet of fuel trucks slamming into buildings, all from the comfort of their own basements.

Do we believe such systems will "freely evolve" without regulation? Will politicians look the other way while these scenarios become a potential reality? Unlikely.

The only question is not if but to which extent they will regulate. As regulation is rarely crafted by engineers

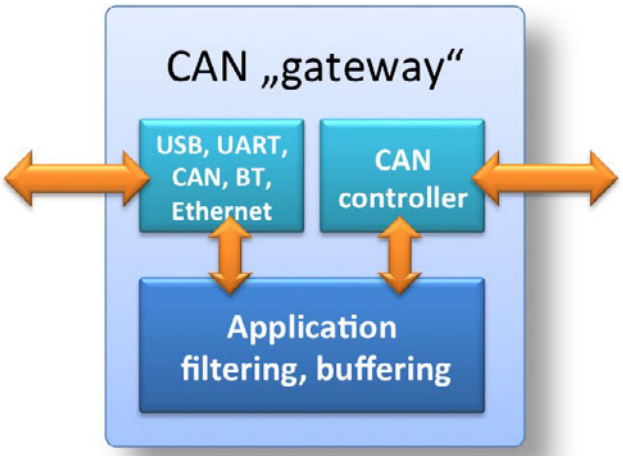

Figure 1: CAN interface, bridge, or gateway with „unlimited“ message forwarding (Photo: EmSA)

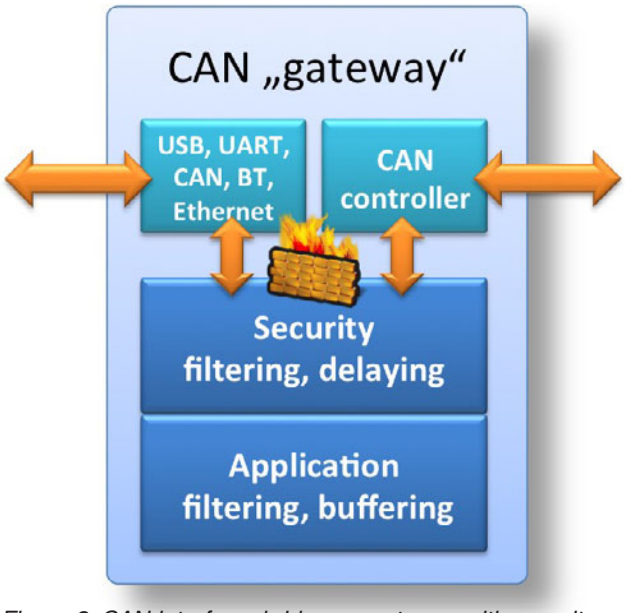

Figure 2: CAN interface, bridge, or gateway with security filtering and bandwidth control (Photo: EmSA)

with in-depth technical insight, we might end up with a law like "state-of-the-art security mechanisms must be implemented at all levels". Great, so also a speed sensor reporting velocity data via some embedded network like CAN suddenly needs state-of-the-art security mechanisms?

Like a sensor reporting wheel pulse counts?

Readers of our last article in this series may remember the challenges in making this signal secure from manipulation. That was only about a single sensor. Now imagine ▷

implementing potentially “security at all levels” for a system with many secure devices. The complexity of the security implementation will grow by an order of magnitude and there are factors such as cost, performance, and power saving requirements that will limit what can reasonably be expected.

But even if all these systems are forced to use the best, state-of-the art security mechanisms, we have to accept that we won’t reach 100 %. Best-effort will mean to only come as close as possible.

Get into the security mindset

For a moment, think of an embedded security application like a house with locked doors and windows as the only protection against unauthorized access. Once a burglar has forced its entry and is inside the house, there are no more protection layers to stop him from also turning on the lights or using the telephone. If a hacker hacks into an embedded system where only the communication to the outside world of one device is protected, they will likely also gain access to all devices that are connected via CAN and be able to manipulate outputs and sensor data. But a house can also have multiple security layers. In an apartment house for example, access to just one apartment does not automatically give you access to the one next to it. There can be alarm systems and valuables can be hidden or locked into safes. Similarly, our future embedded systems will require security at multiple levels – within, among, and between components.

However, it still depends on the specific application how much security is needed for a particular functionality. A subnetwork in the seat of a passenger vehicle to control comfort functions like position or seat heating won’t require the same level of security as the active steering component. The multiple networks in these vehicles are a good example to illustrate what we must change in our mindset for future, more secure applications. Bigger vehicles use multiple busses, some will be based on CAN (FD), others could be using LIN or some Ethernet variant. Crucial for the separation of various level of security are the interfaces, bridges, and gateways between the networks. In the past, the design for all these focused on high performance and throughput as well as reusability and flexibility. Often that meant generic, unlimited access and unlimited and transparent communication between the networks as illustrated by Figure 1.

A CAN (FD) controller or interface can typically produce any CAN message at any rate. Therefore, it can be used in some denial-of-service (DOS) style attack by producing a high-priority message back to back. Same is true for many bridges and gateways, as they are built for performance and generic use they can also pass on “unwanted” or even “dangerous” communication. This needs to change: every CAN controller, interface, bridge, and gateway needs to have some firewall component. A gateway that connects a media/entertainment/information bus to an active steering and control bus should never allow the media side to generate commands for the active steering bus. A hacker with access to the media side should therefore always face a dead end.

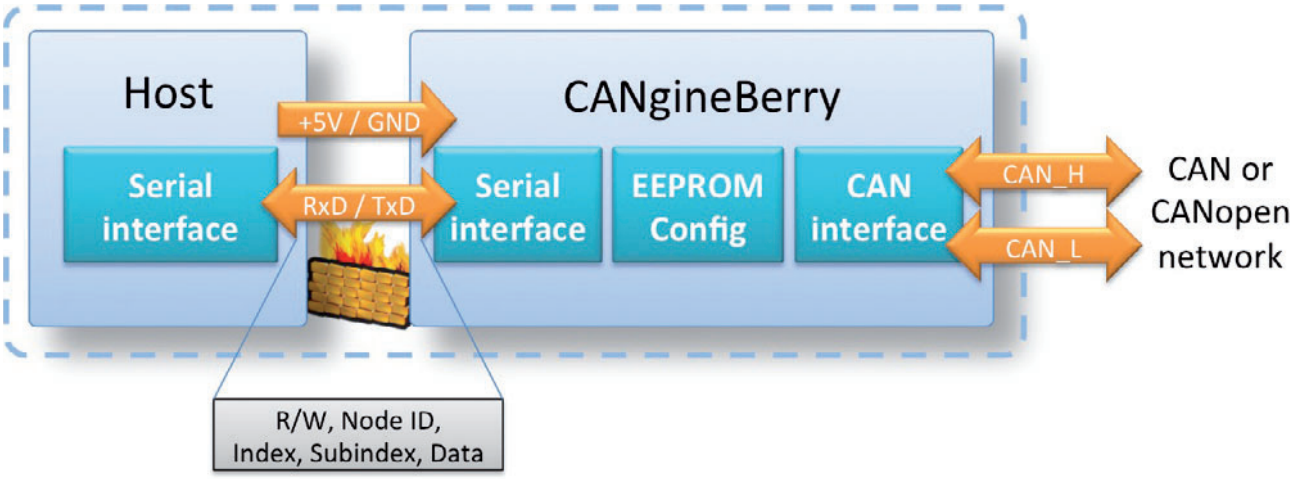

Figure 3: Physical separation: host has no CAN level access, only application data access (Photo: EmSA)

Adding firewall components

The way CAN is currently used, any component on the network can potentially generate any message at any rate. Without further protection levels, a hacked CAN device can be used to flood a network with reset commands or to send very specific control commands. NXP recently introduced a line of smart security transceivers that partially solve this problem in hardware. Once such a transceiver is configured, it acts like a firewall and does not allow "unknown" messages to be produced. However, a hijacked device can still flood the network with allowed CAN messages and seriously limit the available bandwidth or perform a denial-of-service attack.

A software solution at the CAN interface firmware level may be more practical for some systems. Imagine the firmware of such an active CAN interface would be smart enough to integrate firewall components as shown in Figure 2. When transmitting CAN messages, it only accepts "allowed" CAN message identifiers and it can also limit the transmit rate to an accepted maximum by, for example, adding a fixed delay between transmitted back-to-back messages. From such a device it would not be possible to produce a message rate occupying 100 % bus load.

To fully bypass such a system, a hacker with remote access would need to reprogram the firmware of the CAN device. A high hurdle, especially if a secure bootloader is used or the bootloader of this device can only be activated with physical access, e.g. by setting a jumper or using a special connector.

For embedded bridges and gateways with one or multiple CAN (FD) interfaces, the firewall component to add can be similar. At the lower driver level there must be both a flood protection and a filter to only allow well defined, known messages to pass. Preferably, this firewall mechanism is logically or even physically separated from the bridge and gateway configuration which is typically more easily reconfigurable and easier to hack.

Where possible, a hardware separation will provide better protection. For CANopen systems, co-processors like the CANgineberry module serve as an excellent firewall. If the communication between a host and the co-processor does not provide CAN-level but only data-level access, then even a host under total control by a hacker cannot inject arbitrary CAN messages or flood the bus.

Although these methods limit the reach of hacks and the damage that hackers can cause across networks, one issue remains. There will always be an authorized method to generate active controls – and if an attacker reaches full access to the system authorized to send control commands, and manages to keep the system intact, then there is not much else we can do on the embedded firmware side. Shielding this part of the application from the outside world as much as possible and using detection mechanisms against tampering will be essential. ◀

Authors

Olaf Pfeiffer
Christian Keydel
EmSA (Embedded Systems Academy)
info@esacademy.com
www.esacademy.de

Decentralised signal detection and processing

WARRANTY 5 years on ifm products

I/O module designed for mobile applications with integrated PLC

The ioControl module can either be used as a configurable I/O CAN slave in a decentralised control system or as a compact PLC in the field.
The high protection rating and robust housing make it suitable for installation in wet and dirty areas of mobile machines. Programmed with CODESYS.
Practical solutions for automation by ifm – close to you!

www.ifm.com/de/en/iocontrol
Phone +49 800 16 16 16 4