Status summary of CAN security specifications

There is still no dedicated CAN security specification or standard released. Nevertheless, there are committed activities ongoing.

In the last CAN Newsletter issue, the Micro CANcrypt concept was introduced, to show you how to add security to Classical CAN systems with limited resources both in terms of memory and performance. There, we used the additional 18 bits of a 29-bit extended CAN ID to add a digital signature. We promised you some hard, real-world numbers for both memory and CPU resources for this solution, however, got side-tracked by pursuing other CANopen FD customer projects. Customer comes first, but once we have adapted Micro CANcrypt to run on a lower-performance micro-controller according to plan and have actually run it, you can expect to see an update in one of the upcoming CAN newsletters.

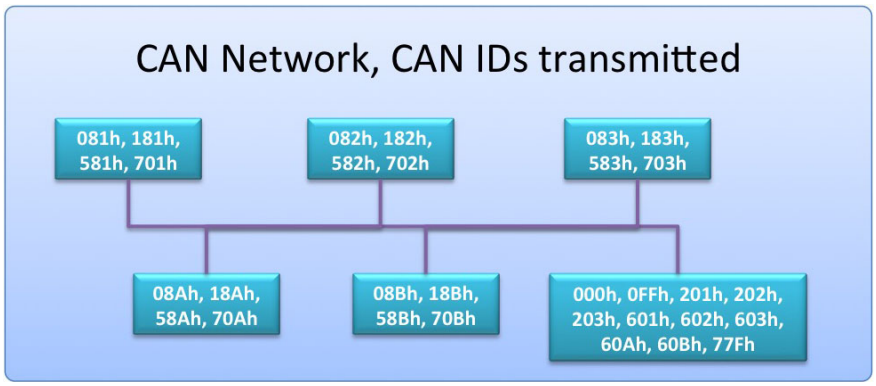

Figure 1: CAN-IDs used for transmissions (Source: Emsa)

End of June 2019, the CiA association hold a phone conference for safety and security issues. Holger Zeltwanger gave the participants an update regarding "base documents". When defining security solutions for Classical CAN, CAN FD, or CAN XL systems, it would be preferable to not start from scratch defining security basics for embedded systems or embedded communication systems. Unfortunately, the current draft of ISO 21434 "Road Vehicles – Cybersecurity engineering" does not seem to be suitable, as it is very generic and not yet completed. It is more of a guideline what designers and developers need to keep in mind when designing a "secured" vehicle. Another document suggested is the "Baseline Security Recommendations for IoT" by the European Union Agency for Cybersecurity. Until the next meeting, CiA will review and report, if that document is suitable to be referred to also by CiA documents.

CAN XL is still in an early specification phase and the related special interest group, recognizing the possibility for security features in hardware to be part of future CAN XL controllers, therefore suggested adding security features to CAN XL first. One of the discussed options is a blacklist/whitelist scheme like the one implemented by the NXP secure CAN transceiver family. Such a scheme can eliminate several potential attack vectors at once if all participants in a CAN (XL) network actively support it. Once we see which security features made it into the CAN XL specification (and hardware), we can review if any of these can still be applied to CAN FD, too, for example on the transceiver level. However, potential CAN controller specific hardware security features will most likely not be suitable to migrate back into CAN FD, so protocol based security solutions are still required.

The essence of blacklist and whitelist handling

In a CAN system the use of the CAN IDs is unique, aside from some very special cases. For each 11-bit CAN ID (or 29-bit when using CAN extended frames) there is only one node in the system, which may transmit a CAN data frame using this CAN ID. Figure 1 shows an example of a simplified CANopen system and the CAN IDs used by each device.

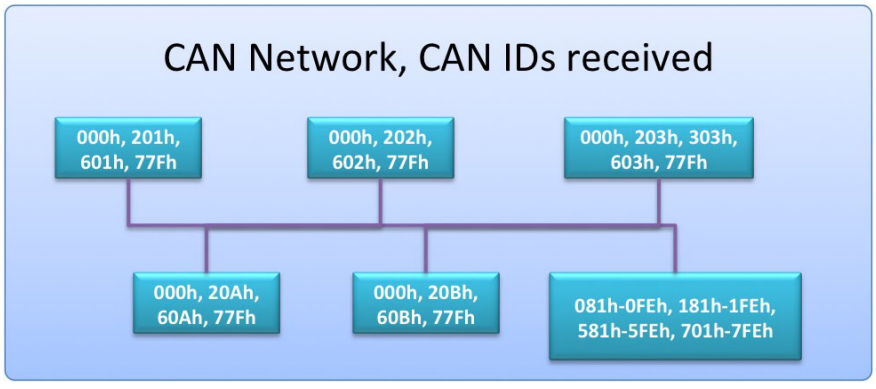

Figure 2: CAN-IDs of received data frames (Source: Emsa)

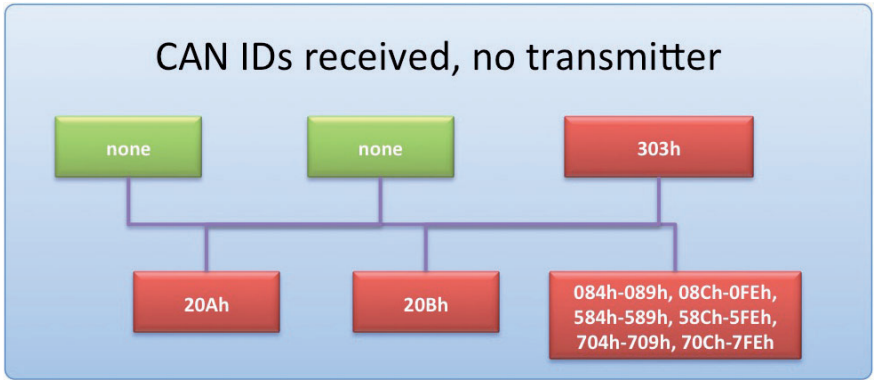

Figure 3: "Unprotected" CAN-IDs (Source: Emsa)

In the scheme, each node applies its known list of CAN IDs used for transmission to a:

- Whitelist: this node may only use the CAN IDs in the list for transmission.
- Blacklist: this node monitors the network to ensure that nobody else uses the CAN IDs in the list for transmission.

This process actively protects the system: If a node detects that a CAN ID from its blacklist is used on the network, it assumes that an attack is happening and someone tries to inject this CAN data frame, for example by using a bus sniffer or a hacked node. If equipped with the ability to generate error frames, the protected node can destroy the injected data frame. The CAN ID is protected to only be used by the node, it is assigned to.

Limitations of blacklist and whitelist handling

As illustrated above, this method ensures that it becomes more difficult to inject CAN data frames. On the receiving side, a CAN data frame with a protected CAN ID can mostly be trusted. Mostly, because there are some attack vectors remaining: if a hacker removes a node either physically or logically by forcing its CAN controller into bus off state, this node no longer protects its CAN IDs. In another scenario, if an attacker hijacks a node, then the attacker can generate any CAN data frames with the CAN IDs that are whitelisted in the node.

Another potential attack vector involves devices, which by default accept CAN data frames from many sources. In CANopen FD for example, devices accept USDO requests from any possible node ID. Not all node IDs will be present in the network, though, so injection attacks using requests assigned to a non-existing node ID will still work.

In order to check a system for remaining injection vulnerabilities, for the whole network you need to verify which CAN IDs are received by the individual devices to determine those CAN IDs that are unprotected. As an example, look at Figure 2, in which all received CAN IDs are shown. Now compare it with Figure 1 that shows all transmit CAN IDs and you will notice that there is a mismatch. This is shown in Figure 3, which has those CAN IDs without a defined transmitter, which are therefore unprotected and still vulnerable to injection.

In this example the device in the bottom right is a CANopen manager where the list of CAN IDs that it can potentially receive is very long.

Even if black and white listing is supported in hardware there can still be serious attack vectors available to intruders, especially if an output device accepts data from many different CAN IDs. Therefore, a true authentication method will still be required for some applications. For CAN XL, the current plan is to have a 1-byte field to indicate if a data frame transports plain data, protocol-specific data or data that includes a digital signature.

Since CANopen FD is poised to support both CAN FD and CAN XL as data link layers anyway, a parameter will be introduced here to globally set the available data size per frame. In addition to supporting the FD and XL versions of CAN, it will potentially also be used to limit the data portion of the frames available to the protocol to make room for a digital signature or other security data. ◀


Authors

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