SMART MOBILITY

PRIVACY AND SECURITY

Making car electronics safe again – a new security architecture for networked embedded devices

Modern vehicles are managed by a network of control processors that interpret sensor readings and operate actuators. These processors control much of the car's behavior and safety functionality, intervening when necessary e.g. for braking, steering, switching on the lights, popping up the airbags, optimizing the powertrain output, and much more. But only fairly recently these networks have also been hooked up to the outside world. This renders them vulnerable to attacks by hackers, a vulnerability for which today there is no effective mitigation available. Jan Tobias Mühlberg, research manager at imec – DistriNet – KU Leuven, explains how researchers at imec have risen to the challenge. The result is a new security architecture for networked embedded devices, carefully designed to fit in today's environments, a solution ready to be used to secure not only smart vehicles, but also other critical infrastructure, e.g. medical equipment, smart buildings, or power grids.

Islands of smart electronics

"Today's complex industrial equipment is monitored and steered by networks of electronics, with sensors, actuators and control processors that continuously exchange messages," says Mühlberg. "In cars, e.g. this interaction is organized around the so-called CAN bus (Controller Area Network), designed as a closed, wired network; an island with no obvious access points for intruders."

The specification of the CAN bus, and thus of networked sensing and computing in vehicles, is about 30 years old. Before, cars were mostly mechanical. The CAN offers a way for the growing number of heterogeneous sensors and control processors in a vehicle to send and receive reliable and timely messages without any sort of central computer. It connects e.g. the rotation sensors in the wheels with the anti-lock braking system (ABS) and with the drivetrain. For the purposes that it was designed for, as a standalone network, CAN works just great.

Jan Tobias Mühlberg: "You'll find comparable networks in industrial control systems and robotic assembly lines. They were all carefully designed and tested to take into account all kinds of exceptional states and errors, which made them quite safe … until recently."

Opening up to the world

Modern high-end cars have infotainment and navigation systems that are hooked up both to the CAN network and to the "outside world". Via these external networks, infotainment components communicate with the driver's mobile phone or headset, and receive software updates from their vendors. And with information from the CAN network, it is e.g. possible to turn up the volume of the music when you start to drive faster, or when you enter rough terrain. Autonomous vehicles will take this a step further and communicate with each other and with the traffic infrastructure to steer the car.

"So suddenly a car's CAN network does have potential entry points for intruders. All this communication with the outside is done over Bluetooth or IP networks, some of which may even connect to the Internet. And the Internet, if anything, is a highly untrusted network", says Mühlberg. "The CAN bus and its hard- and software components were not designed to operate in such an unsafe environment. CAN, for example, has no real form of authentication or authorization. If a syntactically correct CAN message arrives at the car's brake system, the brakes just assume that the message is legitimate and comes from a trusted source, not from somewhere else."

Moreover, the processors are designed to be very small, good enough for their task, inexpensive and consuming as little power as possible. They may run tiny operating systems and a communication and control application. But in contrast to, e.g., laptop or smartphone processors, they don't have memory protection or an isolated sandbox to run processes in. Every application running on a processor, also an application that shouldn't be there, is able to access and rewrite the complete processor memory."

Where is the risk in all this? Mühlberg: "Recently, researchers have demonstrated that they can remotely control a car by hacking its Wifi or Bluetooth gateway. In a high-stakes case in Ukraine, it was demonstrated that electricity grids may be taken over. And researchers at imec – COSIC – KU Leuven even demonstrated that they could hack pacemakers, eavesdropping on the devices and even injecting potentially fatal commands."

This is not to say that such attacks are easy: They require a high level of sophistication, ingenuity and patience. But because of the sheer number of, e.g., electronically identical cars, an attacker that manages to find a way into one system, poses a real threat to the security of very many such systems."

Creating isolated, safe harbors for processing

Today, there is no commercial mitigation available. In contrast to higher-end processors in e.g. laptops and smartphones, controller chips are small and resource-constrained. They lack the security features that have become standard on other processors, such as privilege levels and memory segmentation. And replacing all embedded processors with high-end systems is not an option, mainly because of high cost, complexity and higher power consumption.

“Therefore, we set ourselves the task of designing a secure architecture from the ground up”, continues Jan Tobias Mühlberg. “An architecture that is suitable to secure today’s embedded systems, such as CAN networks in cars, industrial control systems in manufacturing, or very small IoT devices. Such a system has to be low on complexity and cost, which is a definite requirement from the industry.”

The researchers took a lightweight microcontroller as basis, and extended its design, adding secure memory management and a crypto unit that is optimized for low-power consumption. The result is a processor that is not much larger and doesn’t consume much more energy (about 6 percent). But it can isolate the critical software, creating a kind of safe harbor for it to run in. Because of this isolation, the software cannot be compromised. Its trusted computing base is restricted to the hardware on which it runs. Barring vulnerabilities in a protected application itself, no software, be it applications or operating system components, running on the same processor or outside processes, can override security checks and read or overwrite the protected runtime state.

Knowing whom to trust

“But even if the processor that controls the brakes of your car can no longer be hacked, it will still obey a brake command that comes from an illegitimate source”, admits Mühlberg. “Therefore, we limited the trusted sources of messages to those that can authenticate as legitimate. Thus a brake command should only come from a trusted processor, which itself cannot be hacked, and from an authenticated software component. That way, a car’s CAN network is made up of small unbreakable applications that mutually authenticate and trust each other.”

And as an embedded system will still be contacted from the outside, e.g. from a software provider that needs to install updates, or from the traffic infrastructure, imec’s specialists have also implemented secure communication and remote attestation. Thus an outside party can send or receive messages to and from a specific software module on a specific node while being sure that it is the correct module (authenticity), that it has not been changed (integrity), and that its status is correct (freshness).

Demo at ITF Belgium and future work

Sancus, as the solution is called, is a security architecture for resource-constrained, extensible networked embedded systems, that can provide remote attestation and strong integrity and authenticity guarantees with a minimal trusted computing base. It consist of the extended microprocessor, the dedicated software to run in the safe harbors and a C compiler that generates Sancus-secured code.

Sancus is an ongoing project, and the researchers from imec’s DistriNet – KU Leuven and COSIC – KU Leuven groups have a number of outstanding issues that they’d like to tackle.

One is ensuring the availability and real-time functioning of the network. “With our innovation, we can guarantee that any messages that arrive in a module are legitimate,” says Mühlberg. “But we cannot yet guarantee that they will arrive. It would still be possible for an attacker to drop messages, which our solution can detect. In most cases this would probably not lead to dangerous situations, as the receiving node would raise an error and halt the system in a safe way. But it is of course inconvenient.”

A second issue has to do with the safe operation of the secure software modules. Without formal design methodology and inherently safe programming languages, these modules are poised to have vulnerabilities that may lead to unsafe circumstances. But because we have managed to isolate small modules of trusted code, it should now also be possible to design these in a more formal, fault-free way.

Mühlberg's team is looking for collaboration opportunities with partners to develop suitable hardware/software solutions that are adapted to their needs: "At the Imec Technology Forum in Antwerp (ITF Belgium, May 16-17), we'll demonstrate Sancus, either in an automotive scenario or as a smart metering solution, another use case where embedded processors need security. It's also an excellent opportunity for any interested companies to come and talk with us. We can discuss in technical detail how we've managed to add tight security to these embedded networks, an issue that will become all the more pressing as smart autonomous cars start to communicate with their surroundings."

Availability and acknowledgements

To ensure that the Sancus results can be verified and reproduced, the hardware design and software of our prototype have been made publicly available. The hardware designs, all source files, as well as binary packages and documentation can be found here.

Sancus has been implemented by imec – DistriNet – KU Leuven and imec – COSIC – KU Leuven, two research groups famed for their work on security. The development has been supported in part by the Intel Lab's University Research Office. It was also partially funded by the Research Fund KU Leuven, by the EU FP7 project NESSoS, and by the Belgian Cybercrime Centre of Excellence (B-CCENTRE).

Biography

Jan Tobias Mühlberg is a research manager at imec – DistriNet – KU Leuven. Before joining this research group, he did research at the University of Bamberg (Germany, until 2011), obtained his Ph.D. from the University of York (UK, 2010) and worked as a researcher at the University of Applied Sciences in Brandenburg (Germany, until 2005), where he obtained his M.Sc. Tobias is active in the fields of software security, and formal verification and validation of software systems, specifically for embedded systems and low-level operating system components. Tobias is particularly interested in security architectures for safety-critical embedded systems and for the Internet of Things.