Future-proof: Your hands-on checklist on long term cybersecurity for current and future microcontrollers

Your hands-on checklist on long term cybersecurity for current and future microcontrollers – Speakers

Dr. Max Hoffmann

Security Manager – Onboard Security
ETAS GmbH

- MSc in Information Security (2017),
  PhD in Hardware Security (2020)
- Joined ETAS in 2020
- Experienced in the security software development lifecycle, from threat modeling, over secure design, to vulnerability management
- External lecturer @ Ruhr University Bochum

Anthony Esteban

Customer Chief Engineer
ETAS GmbH

- M.Sc. Business Engineering
- Joined ETAS in 2018
- Over 10 years experiences in technical sales for automotive software
- Cybersecurity and Rust ambassador at ETAS

The Challenges with Long-Term Security

Future-proof: Your hands-on checklist on long term cybersecurity

The Challenges with Long-Term Security

Current systems are built for the current market. Foreseeing the future is impossible.

Technological advancements and new business needs may require a change in security approaches.

External factors

– New attack vectors being discovered, e.g., microarchitectural attacks Spectre/Meltdown

– Cryptographic advancements, e.g., Post-Quantum Cryptography (PQC)

Internal factors

– New system architectures, e.g., transition to vehicle computer architecture with HW-supported virtualization, shared memories, new hardware accelerators, multicore applications

– New business models, e.g., monetizing software-locked functionalities

The Challenges with Long-Term Security

Current systems → current market

Future may require a change in security approaches.

External factors

– Cryptographic advancements (PQC)
– New attack vectors → microarchitectural attacks (Spectre/Meltdown)

Internal factors

– New architectures → vehicle computer + HW virtualization, shared memories, hardware accelerators, multicore applications
– New business models → monetizing software-locked functionalities

~~Security-by-Checklist~~

Future-proof: Your hands-on checklist on long term cybersecurity

~~Security-by-Checklist~~

Following a detailed checklist does not lead to a secure system.

Checklists are designed to verify intended steps/results/sequences.
Security is not about what is intended, but about what is possible.

Expert advice:

✓ Don’t rely on a checklist of, for example, testing steps or tools to run, to “prove” security.

✓ Complying with cybersecurity standards, e.g., ISO/SAE 21434, is a good first step, but does not automatically result in a secure product.

✓ Whereas checklists are typically utilized at the end of an activity, consider frontloading cybersecurity activities.

~~Security-by-Checklist~~

Detailed checklist → not guaranteed a secure system

Checklist = verify intention

Security is about what is possible.

- Complying with standards → good first step   BUT   not automatically secure

- Checklists typically at the end -- security from the beginning

Security Mindset Checklist

Future-proof: Your hands-on checklist on long term cybersecurity

Security Mindset Checklist

A better checklist is to verify whether an organization is approaching cybersecurity with the correct mindset.

Security is everyone’s responsibility, not just for “security architects”, “security managers”, etc.

<u>Expert advice:</u>

✓Focus on fostering a culture of **security awareness**.

✓Security engineering being voiced as annoying or holding features back is a **massive red flag**.

✓Consider not just what is voiced but also how people approach everyday tasks.

✓A good indication of being on track is when security is **regularly brought up by different stakeholders and in different contexts**, e.g., safety (ASIL), and everybody discusses seriously.

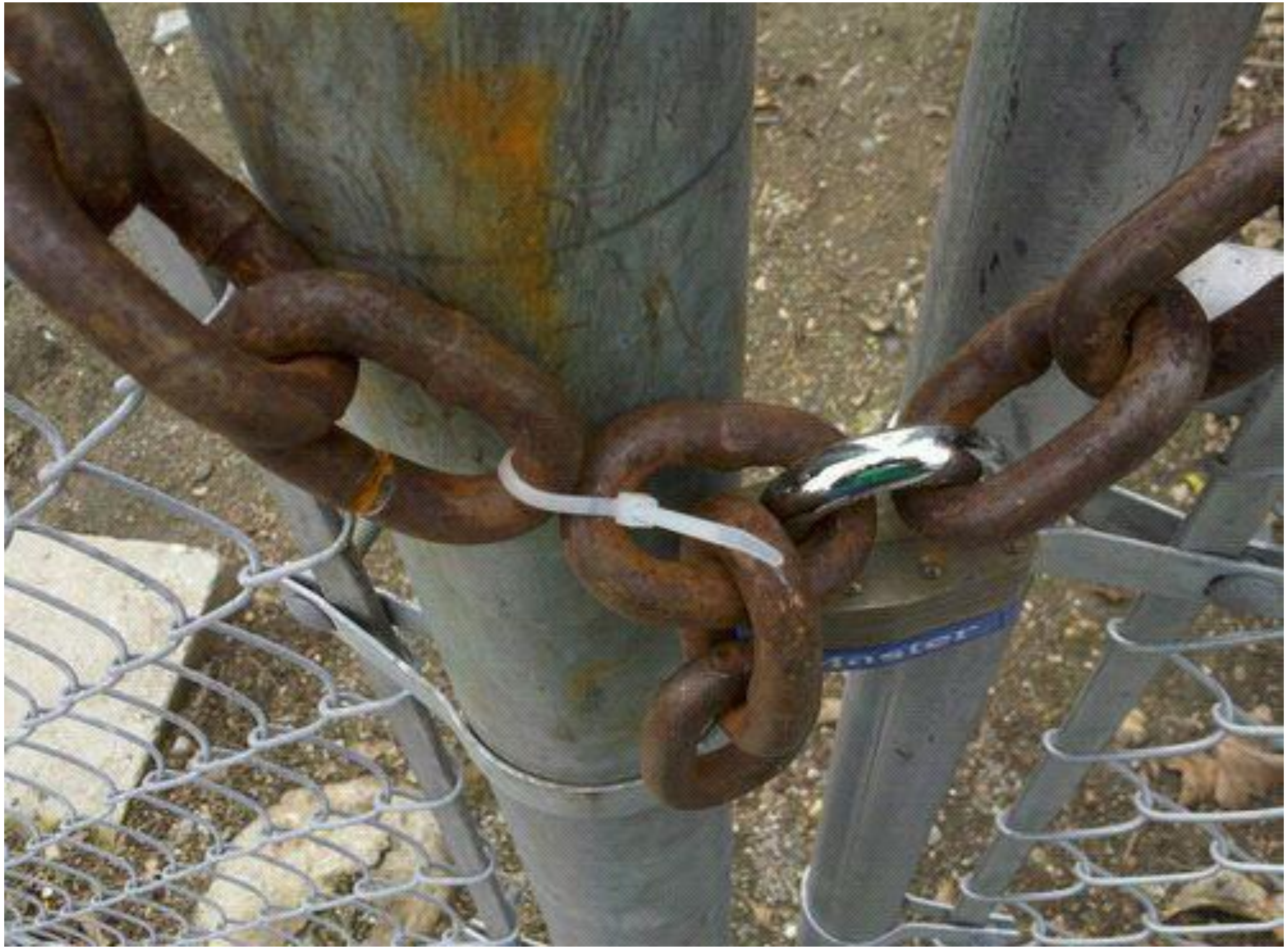
Security Mindset Checklist

Better: checklist for correct approach / mindset.

Security → everyone’s responsibility

- Security awareness
- Security = annoying? Bottleneck? → red flag
- Good: regularly discussion - serious - different stakeholders - different contexts (ASIL)

- We brought a 5-point “Security Mindset” Checklist



Via https://www.reddit.com/r/MemeTemplatesOfficial/comments/qiuzbd/zip_tie_holding_large_chains_and_a_lock_together/
Original source unknown

Security Mindset Checklist #1

The Attacker Model

Future-proof: Your hands-on checklist on long term cybersecurity

Security Mindset Checklist #1: The Attacker Model

Arguing about security is meaningless without a properly defined attacker model.

Expert advice:

✓Analyze possible interactions with your system, **even those that are not intended**.

✓Get inspired by published attacks on similar systems. Collaborative workshops pay into security awareness.

✓Take care not to get lost in detail. The attacker model describes high-level capabilities.

✓It is a valid strategy to assume a slightly improbable worst case. For example, while an attacker never knows all internal details of a product, securing it against such an attacker results in a better overall security design.

✓Ensure that the attacker model is understood by everyone, from management to engineering.

✓Make attacker model considerations visible to customers that integrate your product.

Checklist #1: The Attacker Model

Arguing about security is meaningless without a properly defined attacker model.

Attacker model = capabilities we want to defend against

- Analyze possible interactions
- Inspiration from published attacks
- Model describes high-level capabilities
- Assume worst case even if slightly unrealistic
- Must be understood by everyone, from management to engineering.
- Make attacker model visible to customers

Security Mindset Checklist #2

Defense in Depth

Future-proof: Your hands-on checklist on long term cybersecurity

Security Mindset Checklist #2: Defense in Depth

Assume that a component will eventually get breached. Design for this scenario.

<u>Expert advice:</u>

✓ Security measures against the main attacker model are just a first layer.

✓ After establishing the first layer, switch the attacker model to assume a component was already breached. Add more security layers based on this.

✓ Every layer further reduces the risk of an attacker moving from an initial entry exploit to a meaningful/harmful exploit of the whole system.

Checklist #2: Defense in Depth

Assume component breach. Design for this.

- Measures against main attacker = first layer
- Switch the attacker model → assume component was breached
- More layers based on this.
- (example: car door + engine locked separately)

- Every layer reduces risk
- More difficult to escalate from initial exploit

Security Mindset Checklist #3

Think Like an Attacker

Future-proof: Your hands-on checklist on long term cybersecurity

Security Mindset Checklist #3: Think Like an Attacker

Designing a good defense by guessing how an attacker would approach is destined to fail.
To become a good defender, you need to know the ways of an attacker.

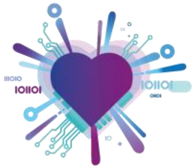
Expert advice:

✓ When it comes to thinking like an attacker, practical experience is key. While a solid theoretical foundation is necessary, it does not teach the creative thinking necessary to carry out hacks.

✓ Setting up a Red Team that tries to exploit the inhouse products is a good approach to build up knowledge and awareness at the same time. Rotating members of the team ensures that everyone is learning.

✓ Capture the Flag challenges (CTFs) and dedicated offensive trainings provide concentrated hands-on experience.

✓ A Threat And Risk Analysis (TARA) provides a helpful structure to explore possible vulnerabilities in a product. TARAs are a great tool to ensure that a product is secure on architectural level.



Checklist #3: Think Like an Attacker

Good defense without taking attacker into account = guessing.
Good attacker → good defender

- practical experience is key
- theoretical foundation necessary, but creative thinking is missing
- Red Teaming, rotate members
- CTFs + offensive trainings → concentrated hands-on experience.

- TARA = good structure to explore possible vulnerabilities

Security Mindset Checklist #4

Embrace Updates

Future-proof: Your hands-on checklist on long term cybersecurity

Security Mindset Checklist #4: Embrace Updates

There is no 100% security. Vulnerabilities do happen. Being able to fix them is vital.

<u>Expert advice:</u>

✓Software AND firmware must be updateable.

✓Updates must be secured, including at least integrity, authenticity, and freshness checks (downgrade protection).

✓Set up development and testing infrastructure, minimize overhead for creating a new release, e.g., via Software Factory

✓Virtualization towards a true SiL setup

✓SBOM

Checklist #4: Embrace Updates

There is no 100% security. Vulnerabilities do happen. Being able to fix them is vital.

- Software AND firmware must be updateable.
- Updates must be secured, including at least integrity, authenticity, and freshness checks (downgrade protection).

- Set up development and testing infrastructure, minimize overhead for creating a new release, e.g., via Software Factory
- Virtualization towards a true SiL setup
- SBOM

Security Mindset Checklist #5

Design for Avoiding Mistakes

Future-proof: Your hands-on checklist on long term cybersecurity

Security Mindset Checklist #5: Design for Avoiding Mistakes

One of the primary root causes for vulnerabilities are mistakes in the software.

Expert advice:

✓ Products are typically designed with the user in mind. For security, we should design architecture and code with the other engineers in mind. A setup that supports engineering to avoid making mistakes greatly improves security.

✓ Practice secure coding, not just to create secure code, but also to create code that is difficult to accidentally misuse.

✓ Consider technologies that are inherently more safe to use, e.g., Rust.

✓ Invest in infrastructure and tooling for early detection, e.g., via fuzz testing.

✓ Every security issue that is caught before reaching production is a small win. Celebrate avoided vulnerabilities for an improved security mindset.

Checklist #5: Design for Avoiding Mistakes

A root cause for vulnerabilities: mistakes in the software.

- Products designed with the user in mind
- Security -> design with other engineers in mind
- Support engineering to avoid making mistakes
  - secure coding → also code that is difficult to accidentally misuse.
  - technologies that are inherently more safe → Rust.
  - infrastructure + tooling for early detection → fuzzing

- security issue caught = win
- Celebrate avoided vulnerabilities → improve mindset

Closing Thoughts

Future-proof: Your hands-on checklist on long term cybersecurity

Closing Thoughts

Security is inherently complex, it is not “finished” at some point and requires constant attention.
But a good security mindset helps to still build a secure product.

<u>Expert advice:</u>

✓Consider security from the beginning, make sure it is understood by everyone.

✓Our security-mindset checklist supports you here:
Start with a clear attacker model, practice defense in depth by thinking like an attacker, be prepared to update your software, and design it to reduce the risks of mistakes reaching production.

✓Keep security awareness high and the security mindset alive. This is an everlasting activity that requires care, a single annual training is not sufficient.

✓Recall that security is never “finished”: take care of security after production, e.g., security monitoring via VSOC.

Closing thoughts

Security is complex, requires constant attention.
Good security mindset helps build secure product.

1. Recap security-mindset:
   - clear attacker model from the beginning, must be understood by everyone
   - practice defense in depth
   - by thinking like an attacker
   - update your software
   - design it to reduce the risks of mistakes
2. Keep awareness high + mindset alive. everlasting activity, requires care.

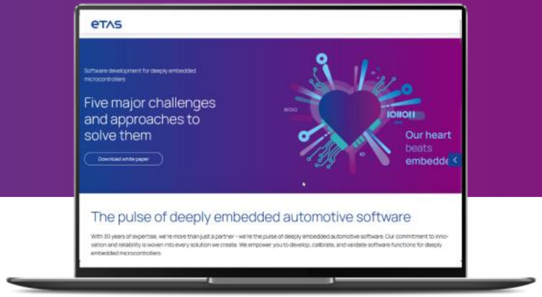
3. Security is never “finished” → post production, e.g., monitoring via VSOC.

Contact us to discuss your needs

Anthony Esteban

Customer Chief Engineer

Contact



ETAS

Software development for deeply embedded microcontrollers

Five major challenges and approaches to solve them

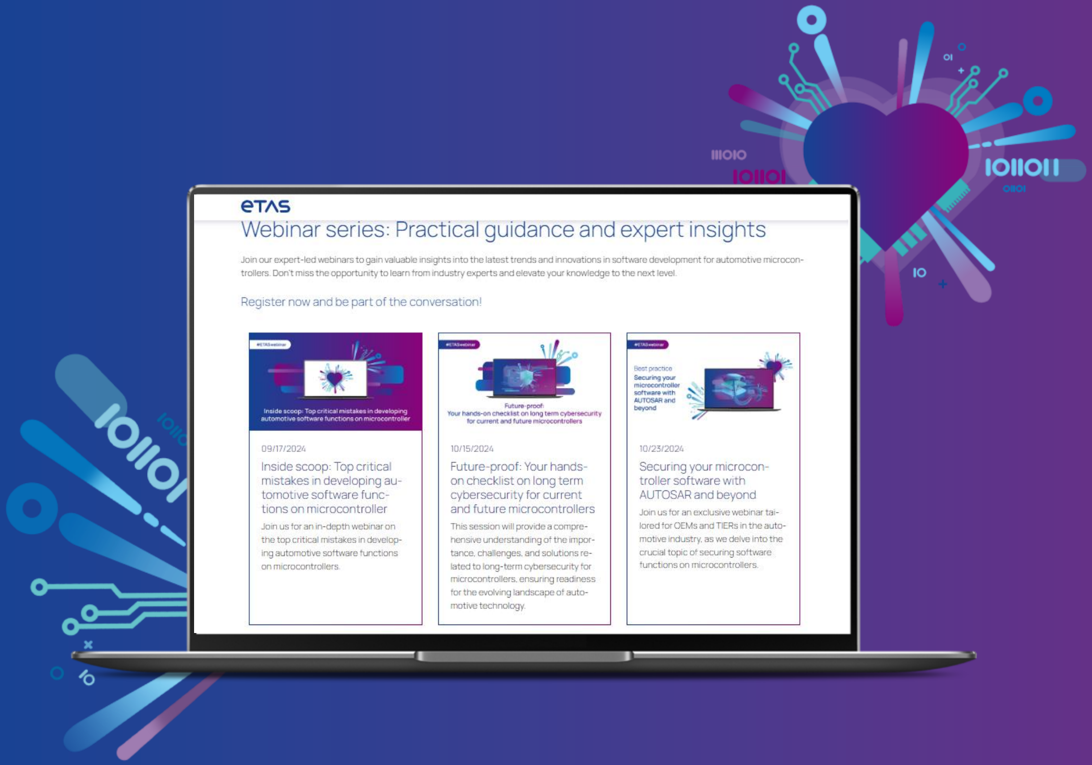
Download white paper

Our heart beats embedde

The pulse of deeply embedded automotive software

With 30 years of expertise, we're more than just a partner – we're the pulse of deeply embedded automotive software. Our commitment to innovation and reliability is woven into every solution we create. We empower you to develop, calibrate, and validate software functions for deeply embedded microcontrollers

Visit our website



Webinar series: Practical guidance and expert insights

Join our expert-led webinars to gain valuable insights into the latest trends and innovations in software development for automotive microcontrollers. Don't miss the opportunity to learn from industry experts and elevate your knowledge to the next level.

Register now and be part of the conversation!

#ETASwebinar
Inside scoop: Top critical mistakes in developing automotive software functions on microcontroller
#ETASwebinar
Future-proof:
Your hands-on checklist on long term cybersecurity for current and future microcontrollers
#ETASwebinar
Best practice
Securing your microcontroller software with AUTOSAR and beyond

09/17/2024

Inside scoop: Top critical mistakes in developing automotive software functions on microcontroller

Join us for an in-depth webinar on the top critical mistakes in developing automotive software functions on microcontrollers.

10/15/2024

Future-proof: Your hands-on checklist on long term cybersecurity for current and future microcontrollers

This session will provide a comprehensive understanding of the importance, challenges, and solutions related to long-term cybersecurity for microcontrollers, ensuring readiness for the evolving landscape of automotive technology.

10/23/2024

Securing your microcontroller software with AUTOSAR and beyond

Join us for an exclusive webinar tailored for OEMs and TIERs in the automotive industry, as we delve into the crucial topic of securing software functions on microcontrollers.

Coming up webinars:

October 23
Securing your microcontroller software with AUTOSAR and beyond – more information

October 29
Mastering fuzz testing: How ETAS and Keysight empower the automotive industry to overcome cybersecurity challenges amid regulatory compliance – more information

November 6
Measurement, calibration & validation for any vehicle at its best – more information

November 26
Opportunities and limits of virtual testing – more information

December 10
Ask the expert: Bring your ECU software development process problem and we discuss – more information

Thank you!