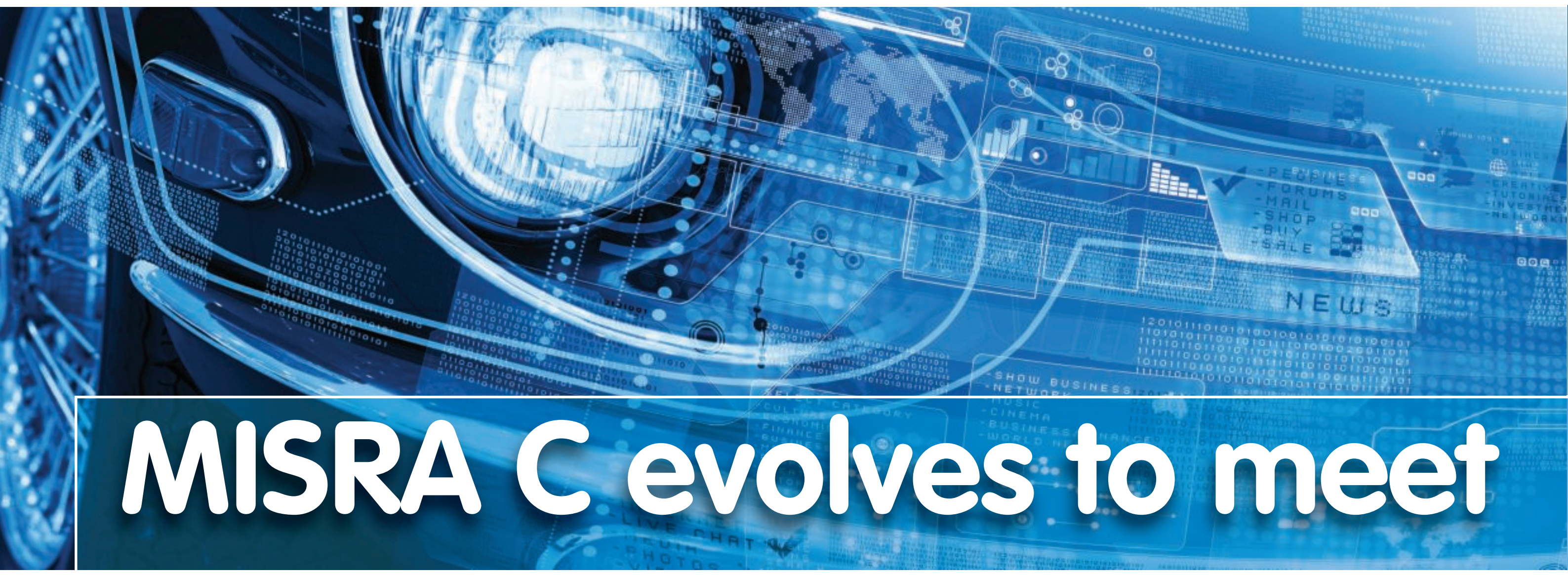


MISRA C evolves to meet

With the increased connectivity of applications and devices, OEMS are demanding new assurances that software has been developed to the highest security, as well as safety, standards.

By **Neil Tyler**.

At a time when security risks are growing on the back of rapid growth in the IoT, there has been an upsurge in the assurance requirements associated with and required for embedded devices. Connectivity is at the heart of the IoT and even devices that were not designed originally to be connected are now being adapted to deliver greater connectivity.

As a result, more products than ever are seen as vulnerable to hackers, as well as to the inadvertent release of data or personal information – which means that coding needs to be secure and coding practices need to be thorough, documented and well understood.

Originally designed as a software development language subset to promote the use of the C programming language in safety critical embedded applications deployed in the automotive space, MISRA C has been updated over the years as the guidelines have evolved to address the needs of safety critical applications, not only in the automotive space but also in mission-critical applications across a broad range of industries.

"If an application is not secure, then it is not safe," explains Jim McElroy, vice president of marketing for LDRA Technology. "And as the embedded software industry has evolved, so too have the threats from hackers."

MISRA, originally created as the Motor Industry Software Reliability Association, comprises a consortium working to promote best practice in developing safety and security related embedded systems.

According to McElroy, the MISRA C guidelines were originally intended to allow programmers to spend 'more time coding and less time on compliance issues'.

Although the MISRA language subset has strong ties to the safety-critical market, it has become a proven approach for best coding practices for any embedded application.

"As the embedded software industry evolves, so have the threats – whether that is to lives, equipment or businesses," explains McElroy. "And while MISRA guidelines are designed to help developers write high quality code, which is by its very nature more safe and secure, increased industry awareness of security risks has led to the need for guidelines to evolve in order to address them."

As developers add features to their products, working with constrained budgets and with increasingly tight schedules, software can become the weak link when it comes to preventing hackers gaining access to sensitive data or taking over systems.

"As a result, security has moved centre stage. It can't be viewed as an afterthought in terms of development," says McElroy. "It has to be designed in from the very beginning using coding best practices and rules designed to protect the safety and security of OEMs and their end users."

In response, the committee with responsibility for maintaining the C Standard published the ISO/IEC 17961:2013 C language Security Guidelines. Following on from that, the MISRA committee has released the MISRA C: 2012 Amendment 1.

Amendment 1

"The amendment has been designed to support a range of new requirements for improved security," McElroy notes.

It specifically establishes 14 new guidelines for secure C coding to improve the coverage of the security concerns highlighted by the ISO C Secure Guidelines, among them a guideline that addresses the specific

new security threats

issue of 'untrustworthy' data – 'a well known security vulnerability', in McElroy's opinion.

Amendment 1 is an enhancement to, and is fully compatible with, the existing MISRA language guidelines and will be the standard approach for all future editions.

"At the heart of the MISRA C:2012 Amendment 1 is the aim of helping developers avoid coding practices that can introduce security vulnerabilities and to write code that is more understandable and maintainable," says McElroy.

"These amendments encourage developers to follow additional guidelines, which will enable developers to more thoroughly analyse their code and as a result be able to assure regulatory authorities that they have followed safe and secure coding practices."

In the past, MISRA C was open to interpretation and the amendment will require greater transparency and for all processes to be thoroughly documented.

McElroy suggests that many people could deviate from the standard while claiming compliance. "Developers were claiming compliance, but were not testing sufficiently, whether that was because of cost or a lack of resource," he suggests, pointing to the experience of smart meter providers who rushed product to market, overlooked security and are now having to address the issue retrospectively. "Security is a critical issue and security threats have led to stringent new requirements by OEMs that require developers to prove that their software meets very highest standards.

"The MISRA standard is now more about compliance and the recent amendment has certainly tightened it up; if a supplier intends to deviate from the standard, it now has to explain to its customers why it intends to do so and in what way."

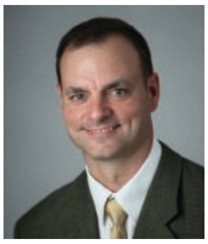
Compliance

So are you compliant in the light of these amendments? Key questions for any developer include: are you using tools that can thoroughly check compliance and then prove it to both customers and regulatory bodies; does the tool provide a full compliance matrix so the developer will be able to see exactly what is being checked; does the tool being used check against other relevant standards (such as CWE and CERT); and can the tool provide dynamic analysis, which means that 'dead code' can be identified and any threats to security addressed?

The new MISRA guidelines mean that developers will have to analyse their code thoroughly and assure regulatory authorities that they are following safe and secure coding practices.

"Developers will have to be able to demonstrate code clarity, consistency and that it complies with the necessary standards," McElroy says.

"As the embedded software industry has evolved, so too have the threats from hackers."
Jim McElroy



For customers in critical industries, OEMs have a long list of stringent requirements for developers.

LDRA's MISRA compliance checking tools, for example, provide developers with an immediate and comprehensive coverage of all documents.

"This not only offers what we believe is the most comprehensive adoption of the MISRA C:2012 standard, but also, thanks to integration within the LDRA tool suite, enables our customers to check against other security or safety standards and any industry specific standards," says McElroy. "Developers will also gain advantages by automating their coding standards compliance within their overall software analysis and testing process.

"There are plenty of examples highlighting what can happen if security isn't taken into consideration or designed into the product from the beginning, along with the risks associated with a breach in security after a product is launched," says McElroy.

"Companies like LDRA are developing automated tools to help developers when it comes to developing high-quality software using best practices and coding rules."

According to McElroy, there's no longer any reason for companies to cut corners when it comes to compliance. "The risks associated with not doing so have become infinitely greater," he concludes.