LDRA coverage
LDRA news from the world of testing
Issue 29 Summer 2024

Chara Technologies: Driving the zero emissions future with the LDRA tool suite

The Client

Chara Technologies' development team, based in Bangalore, India, is focused on the development of Rare-Earth free and affordable motors. The emergence of electric and hybrid vehicles in recent years has led to heightened interest in Chara's Synchronous Reluctance Motors (SynRM). These can typically develop higher continuous torque levels than similarly sized induction motors.

The development work for Chara's SynRM range includes proprietary motor control algorithms, communication stacks, application layer software and a cloud-based analytics platform handling motor data.

The Project

Omkar Lad, Head of Firmware & Software at Chara Technologies, takes up the story.

"We are currently in the process of developing products catering to the global markets for 2, 3, and 4-wheeled vehicles. A majority of 4-wheeler Original Equipment Manufacturers (OEMs) have mandated compliance with ISO 26262 and ASPICE standards, and the others are progressively adopting them. Some off-road and agricultural applications require us to align with ISO 25119. All these process standards necessitate the structured development of software."

The Decision

To address the challenges posed by the obligation to adhere to these standards and guidelines, Chara Technologies settled on the LDRA tool suite. They were impressed not only by its integrated static and dynamic capabilities, but less obviously by its intuitive, easy-to-use interface.

The LDRA tool suite has yielded significant commercial advantage for the company. "Since using the LDRA tool suite, we have saved the equivalent of a developer's time on code review." said Omkar. "By replacing the open-source unit test framework we relied on previously, we have gained reporting on code coverage, test results, and control & data flow, all of which are needed for certification. The tool suite makes the certification process much smoother."

Using the LDRA tool suite has resulted in other, less obvious benefits for the development team. In particular, Omkar noted that they have adopted complexity and other code quality metrics since using the tool, and that their code quality in general has improved significantly – perhaps with the benefit of reduced support overheads in the future.

Broadening the impact

Chara Technologies are so pleased with the LDRA tool suite that they apply it even where standards are less demanding. "Outside the automotive sector, many of our projects need not comply with such stringent functional safety requirements. Even so, the LDRA tool suite allows us to apply the same rigorous verification and validation processes, ensuring higher reliability and quality for industrial and HVAC customers."

"We confidently predict a return-on-investment period for the LDRA tool suite of no more than 18 months" Omkar concluded. "Even the accountants are happy."

Read an extended version of this story at

Blog Page

Inside this issue

Taint analysis: Why should you care?

Like it or not, connectivity is a big deal in the world of embedded systems. It has opened a whole new world of possibility, through enhances communication and collaboration, remote monitoring and management data exchange and analysis, over-the-air updates – the list is almost endless.

"To share your weakness is to make yourself vulnerable; to make yourself vulnerable is to show your strength"

Criss Jami

But there are serious downsides too, and collectively they have a significant impact for embedded system developers.

Increased connectivity exposes critical embedded systems to cybersecurity threats. As these systems become more interconnected, they become potential targets for cyberattacks. Securing critical systems against unauthorized access and ensuring data integrity become paramount concerns. It is no coincidence that cybersecurity standards pertinent to embedded systems are becoming ever more prevalent. Examples such as DO-356A (civil aviation), ISO/SAE 21434 (automotive), and ISA/IEC 62443 (industrial automation) represent the tip of the security standards iceberg – and almost without exception, taint analysis is a recommended practice in every case.

Tainted data

Follow any application inside a debugger and it is easy to see that information is being copied and modified all the time. Taint analysis is focused on data that is input into this melting pot of moving data from an "untrusted source" – that is, from a source that cannot be verified within the scope of the recipient application.

Unhappily, the use of the term "tainted data" varies, with some commentators using it to reference anything from such a source, and others using it to only reference data that is from an untrusted source and has been compromised in some way. Here, we use the former definition.

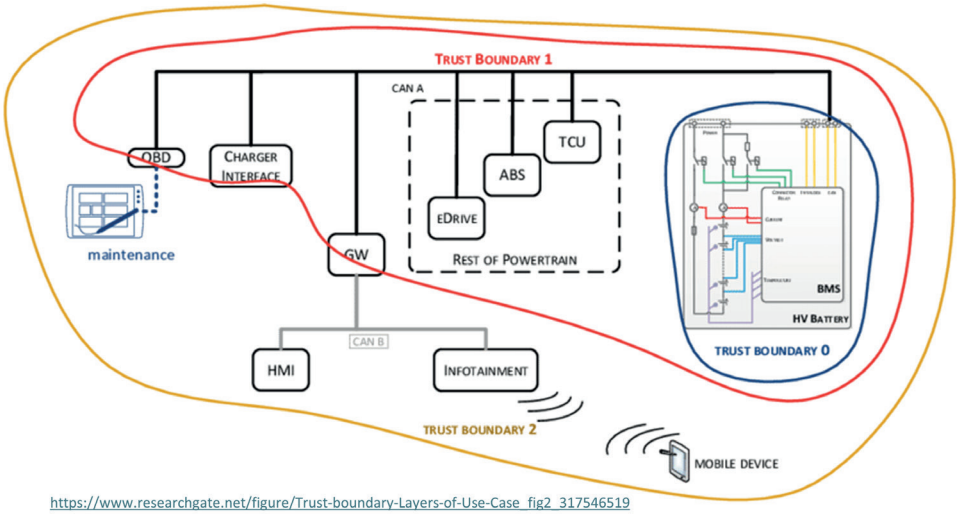
Vulnerability analysis

A "shift left" approach to taint analysis as promoted by our example standards begins with Software Vulnerability Analysis (SVA) at design time. The illustration below shows a schematic of how that might be achieved using Trust Boundary Analysis (TBA) on the design for an Electric Vehicle (EV) in accordance with ISO/SAE 21434 §8.5.

In the Security Vulnerability Analysis (SVA) process, the initial step includes decomposing the application, analysing data and control entry/exit points. Controls are established at trust boundaries, documented in a threat model using specialized data flow diagrams. This guides the design team in prioritizing cybersecurity validation efforts, such as following secure coding practices like data sanitization. The second step involves threat categorization using systems like Common Vulnerability Scoring System (CVSS) and Common Weakness Scoring System (CWSS) associated with CVE and CWE respectively, aiding in identifying vulnerabilities at various software levels.

"I give you all aboy could give you
Take my tears and that's not nearly all"

Edward Cobb



https://www.researchgate.net/figure/Trust-boundary-Layers-of-Use-Case_fig2_317546519

Trust boundary identification is a common approach to software vulnerability analysis (SVA)

Taint analysis: Fast forward to development time...

Taint analysis involves marking data as "tainted" when it originates from such sources and then tracking how this tainted data propagates through the program's execution flow. By providing a comprehensive understanding of how data moves through a program, it enables developers to pinpoint areas of concern and implement targeted security controls.

Ideally and in accordance with the "shift left" paradigm, that analysis should take place with reference to the earlier SVA - although development time taint analysis does not directly depend on that.

> ***"Any plan with more than four steps is not a plan, it is wishful thinking."***
> *General Wesley Clark*

Taint analysis can be achieved in four steps using the LDRA tool suite.

Tainting data: Mark data as "tainted" when it originates from an untrusted or external source such that the data that could be influenced by an attacker.

Communications with the "outside world" – user input, serial communications, ethernet – are obvious examples of tainted data.

Less obvious examples include POSIX or other RTOS calls, access to shared memory between hypervisor partitions, and inter-partition communication between RTOS partitions including TCP/IP communications.

Taint analysis in the LDRA tool suite is configurable to identify pertinent keywords to account for all such examples, and more. Illustrations of the reports are shown below.

Tracking Data Flow: Trace the flow of tainted data as it moves through a program or system. This involves tracking how the tainted data is manipulated, processed, and used within the code. The LDRA tool suite leverages call diagrams (call graphs) to show this statically, supplemented by unit test and/or robustness tests to input data via each source to assess how it is handled.

LDRA tools do not analyse only what code might do based on static analysis. They can show what it will do when it is compiled and run on the environment in which it will operate in the field. What's more, they can demonstrate HOW rogue tainted data is handled and confirm that the mechanisms surrounding that are functional and effective. No static analysis tool alone can do that.

Identifying potential security vulnerabilities: For example, if tainted data is used in a database query without proper sanitization, it could lead to a SQL injection vulnerability. If tainted data is leaked to the user interface, it could result in sensitive information disclosure. Identifying these vulnerabilities is about ensuring the code is good quality – that is, the usual LDRA static analysis approach applies. Coding standards including CERT, MISRA and CWE are pertinent, as is code complexity.

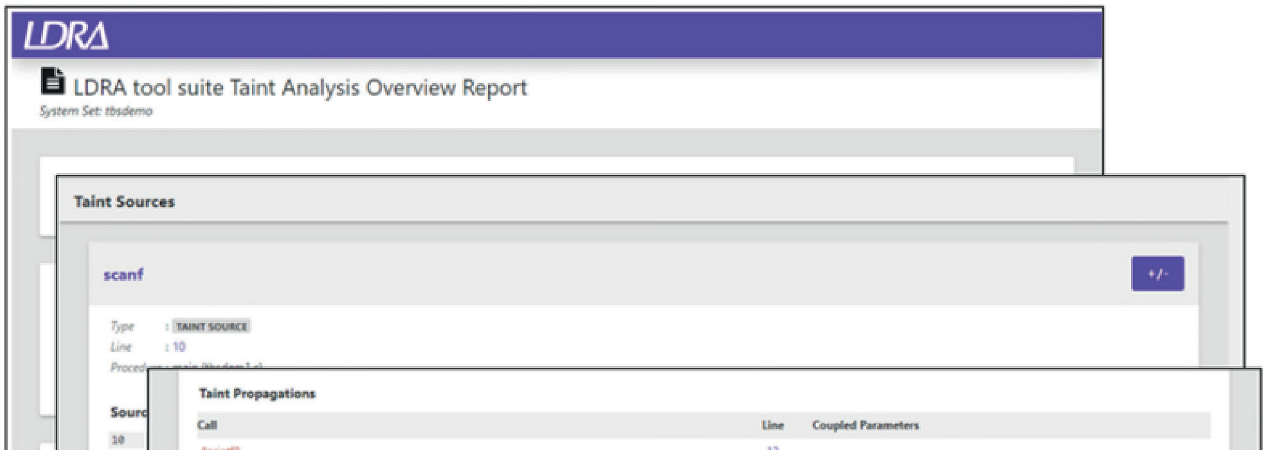
Reporting and mitigation: Identify vulnerabilities to developers so that they can take appropriate measures to mitigate the risks. This may involve fixing the code, adding input validation, or improving data sanitization procedures.

In summary

Taint analysis enhances software security by tracking and mitigating the flow of untrusted or sensitive data. It aids in identifying and preventing vulnerabilities, strengthens threat models, and supports prioritized cybersecurity validation. Taint analysis aligns with secure coding practices, contributing to the creation of more resilient and secure software systems.

If you'd like more information on how LDRA can help with your safety-and security-critical developments, please do get in touch. Details on the next page.

> ***"Communication is your ticket to success if you pay attention and learn to do it effectively***
> Theo Gold

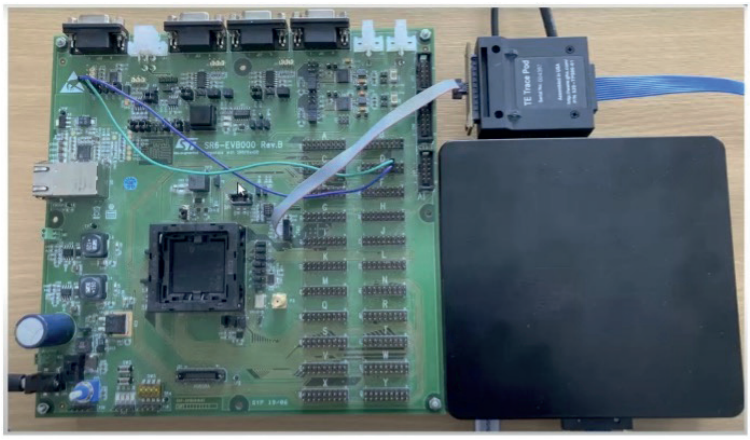


Tool Integration News

As part of LDRA's policy of continuous improvement we have an ongoing programme of integration with hardware and RTOS,

Recent hardware examples include the ST Stellar SR6 P7 (below). This is an example of the Stellar integration MCUs which have been designed to meet the requirements of domain controllers and ECUs with high integration requested in the architectures of connected update-able automated and electrified cars. You can learn more via https://www.youtube.com/watch?v=hyuTTdyYjsY

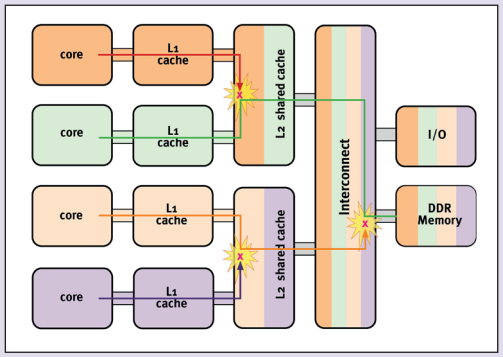
Our second example is the integration of the LDRA tool suite with QNX SDP 8.0, the next-generation Software Development Platform from QNX. According to QNX it is their "foundational development platform for the next generation of mission, safety- systems—merging unprecedented performance with unparalleled security and reliability—without compromise".



LDRA Newsroom

AC 20-193 – Use of Multi-Core Processors

On 8th January 2024, the FAA officially issued "AC 20-193 – Use of Multi-Core Processors", complementing the earlier EASA release of AMC 20-193. The primary MCP challenge concerns the estimation of Worst-Case Execution Times in view of interference channels resulting from shared resources.



A combination of static and dynamic analysis presents the most pragmatic approach to achieving the execution time assurances demanded across the safety-critical sectors. It will detail a flexible wrapper-based technique that fulfils the A(M)C 20-193 civil aviation objectives. Consultancy services are of course available to support you in achieving those objectives, but equally the TBwcet can be licensed in the usual way if you have no need for them.

Read more about the general case here: https://ldra.com/capabilities/mcp/, and of the implications for civil aviation here: https://ldra.com/amc20-193/

Social Networking

Join us on our Twitter, Facebook and LinkedIn profiles to get the latest updates about LDRA news, product updates, events and webinars.

@ldra_technology
LDRA Software Technology
LDRA Limited

LDRA Software Technology

Dates for your diary for 2024

Training	
Understanding MISRA C:2023	**1 – 2 Oct**
LDRA tool suite training course	**22 – 24 Oct**
Webinars	
How Long is a Piece of String? Complexity and How to Measure it	**20 June**
Leveraging Model Driven Software Development and Automated Software Verification to Deliver Next Generation Avionic Systems	**9 July**
Conferences & Exhibitions	
MOSA Industry and Government Summit 2024	**17 – 18 June**
ESSS 2024	**16 July**
ESSS 2024	**18 July**
ESSS 2024	**23 July**
ESSS 2024	**25 July**

Visit the LDRA events page

LDRA

Tel EMEA: +44 (0)151 649 9300
Tel USA: +1 (855) 855 5372
Tel India: +91 80 4080 8707

Email: info@ldra.com
Web: www.ldra.com

Newsletter Contributions

Contributions from our readers are welcome. If you have any comments or stories that you feel are relevant to the world of software testing please contact us.