

Secure, Open, Real-Time Operating Systems

Submitted to: Networked Embedded Control for Cyber Physical Systems (HCSS-NEC4CPS)

Authors: Brian Isle, Todd Carpenter, Charles Payne, and Kyle Nelson, Adventium Labs

Mr. Brian Isle, Chief of Operations and Member of Technical Staff, brian.isle@adventiumlabs.org 612 716 5604

The following thoughts apply to several of the workshop topics including Certification Issues, Methods, and Systems Issues in the context of secure real-time operating systems (SRTOS).

Background: The term “computer security” has morphed from proactively assuring information protection to reactively defending against cyber-attacks. This is due to several contributing factors including:

- 1) The acknowledged difficulty of evaluating real systems against the Trusted Computer System Evaluation Criteria (TCSEC) or “Orange Book”.
- 2) The military and commercial worlds embraced different models of security, so solutions that satisfied the military could not easily be transitioned to the commercial sector. The military was most concerned about data confidentiality (e.g., use of sensitivity labels like Confidential, Secret, Top Secret), whereas the commercial world was more concerned with data integrity and availability, often driven by safety and financial considerations.
- 3) The emergence of the PC in the 1980s was a very disruptive technology for the security community. Once computing was so easily available on the desktop, everyone wanted that capability and the packaged applications that came with it. Unfortunately, this technology was not designed from the ground up with security in mind, and we have spent the last two decades being reminded of that fact.

Safety and security challenges continue and the world continues to become more complicated; today’s systems are far more networked than those considered when the core science was developed. How do we interpret a security kernel, reference monitor or trusted computing base now, and yet still support the safety requirements, such as guaranteed operator access when dealing with emergency situations? Pervasive computing, ubiquitous computing, wireless access, MANETs, etc., all challenge the traditional way of thinking about computer systems. Composing computer security requirements with other foundational requirements, such as real-time, cannot proceed until we fully understand each set of requirements in isolation.

Recommendations for addressing SRTOS

We propose an integrated roadmap to address these challenges to enable distributed, real-time, embedded systems that in the future incorporate security and assurance. The roadmap can be segmented into the following categories:

Fundamental Security Science: An established need, significant resources, and great talent have not been enough: we still use old technology to thwart ancient threats. Currently, very specific real-time, fault-tolerant solutions exist in narrow domains and we can reasonably communicate on matters of real-time and fault-tolerance. Tools are lacking, however, and address only specific parts of the problem, with no known integrated, end-to-end environments. The various abstractions address portions of the problem, but leave resolution of inter-model gaps to the humans. The limitations of ‘testing’ are not well communicated. We continue to rely on them, because that’s what we have. To make progress, a clear and precise language for security requirements is required, the physical, hardware, software, social domains (an SRTOS does not exist in a vacuum) all need to be addressed, and tools and methods for secure, real-time, fault-tolerant, mixed criticality and multi-level security development and analysis are needed, without these tools industry cannot broadly incorporate it.

Threat and Failure Models: Industry lacks the incentive, awareness, and ability to do meaningful tradeoffs, despite availability of comprehensive techniques. Currently, it is remarkably difficult to justify applying technology that is already available, and it is nearly impossible to justify future technology. Companies make decisions based on known costs, which are hard to garner when everyone is hiding their issues. We have all heard “If I know about a problem I have to fix it.” and “Why would they ever want to attack us?” Unless there is economic or regulatory incentive, why add security? Staving off some theoretical future maybe-threat is not financially responsible to the shareholders. To make progress, it is necessary to develop domain specific attack and defense taxonomies (more expressive and complete language to describe the issues and to make sure we get coverage) and threat and failure modeling methods, tools, testbeds, and information repositories that companies can use to identify

and quantify risks at a sufficient level of detail to permit tradeoff of options based on real risk. Threat models need to include attacker perspectives and provide explicit mapping to defender resources.

Integrated Approach: Integrated models, abstractions, approaches, and capabilities are lacking; instead there are a plethora of diverse formal methods and specification approaches. Formal methods are tried, evaluated, and used every now and again, but scalability is a real issue – the tools and techniques work in very limited areas which are insufficient to address the whole problem, so we end up with multiple models, and humans have to resolve the discrepancies, usually by test. Which misses things. To make progress, it is necessary to integrate various security techniques - get world class NSA expertise transitioned both to industry and academic capability. In addition, need to address physical/hardware (e.g., Trusted Platform Module) /software/application integration. The whole problem must be addressed, including real-time, fault-tolerant capabilities, and all the other critical ‘ilities.’ Finally, lifecycle tools and techniques must cover specification, development, certification, use, and maintenance/evolution.

Regulation and Liability: Industry is (immediate) cost driven. Regulation helps, but that will obviously be a nasty political battle. Regulation that is meaningless drivel is possibly worse than none at all, because it diverts resources to nonproductive ends. Currently, there are dramatic differences between regulated industries vs unregulated. Consider avionics vs process control. The notion of ‘real-time’ and ‘fault tolerance’ is wildly varying, though many of the techniques are shared. In critical infrastructure – regulated industries – it is tricky to force updates, though they’re clearly needed. Yet those industries can pass real costs down to consumers. Some current regulation is stronger than others and (unsurprisingly) has a lot of good effect. It might stifle some innovation and penny players, but, for instance, aviation is incredibly safe. Compare that to, say, ecommerce and the raging identity theft problem. As a case in point, one of today’s bigger motivators today for security is the California law which requires companies to notify users of inadvertent information disclosures, though that law is steadily being eroded and weakened by the same people who lose your information. To make progress, research in regulatory and liability approaches is needed to motivate acceptance and responsibility. Instead of opinions, we need business research – what does motivate companies? How would you find out? How would you be able to learn it without those same companies blocking you with politics, lobbying, and special interest groups?

In conclusion, we (as a community) know where to start. In fact we know more than we sometimes appreciate about building secure systems. To make progress however, significant *cooperative* progress needs to be made by all stakeholders (Academia, Government, and Industry) to 1) research basic science foundations necessary for security, safety, real-time, fault-tolerance, etc., 2) develop open systems built ‘from the ground up’ with the key attributes, and 3) provide enabling tools and techniques for construction, analysis, and certification of such systems.

Company description

Founded in 2002, Adventium Labs is a non-profit research and development lab, focusing on the development of advanced software applications for complex systems, with a particular emphasis on automated reasoning, human-system interaction, and supporting architectures. Adventium Labs is dedicated to performing and publishing scientific research and to the creation, maturation, and commercialization of intellectual property. Our goals are to advance the state of the art, to provide effective solutions to important problems, and foster the creation of high-tech jobs and companies. Our Adventium team has worked directly on related systems, considering technology, safety, security, regulatory, maintenance and support, usability, and business issues. This provides grounded experience for what it takes to develop and apply an OS, RTOS, or SRTOS in real-world, life and safety critical applications.

Author Biography

Brian Isle (workshop attendee) is the chief of operations and has been a member of the technical staff at Adventium Labs since its inception in 2002. His current technical focus is in critical infrastructure safety and security. Mr. Isle is a key member of the Air Force Association’s Critical Infrastructure Team, which is responsible for creating the threat scenario-based Vulnerability Assessment and Prioritization Methodology and the application to Minnesota’s homeland security effort. He recently supported the onsite demonstration of a next-generation embedded firewall technology at the Joint Warrior Interoperability Demonstration ’04 (JWID04) at U.S. Northern Command (NORTHCOM). Mr. Isle is currently supporting a DARPA program developing approaches to automating aspects of vulnerability assessment for force protection at military bases and a DHS program to apply advanced distributed firewall technology to SCADA applications. Mr. Isle is the chair person for the PCSF special interest group on cyber self assessment for manufacturing and process control systems.