SAFETY-CRITICAL VERSUS SECURITY-CRITICAL SOFTWARE

Findings in this area from research work with leading Professionals and Academics and in conjunction with BCS, Resource Engineering Projects and SCSC

Dr Adele-Louise Carter

TD, MBA, BSc (Hons), FBCS, CITP, CENG, MCMI, CMgr

Fellow of the BCS, REP Software Sales Manager and member of the SCSC

adelelouise.carter@bcs.org.uk

07525 716964

Version 1.0

August 2010

1. Introduction

In the past, safety and security-critical software systems may have been considered completely separate due to the differences between 'safety-critical' and 'security-critical', with the latter having the added implication of loss of human life.

The functions of both types of digital systems have become increasingly software intensive and this has contributed to a blurring of the difference between safety and security-critical. For example, the fault on an aircraft's flight control function could lead to a catastrophic failure condition, which results in the loss of human life and hence the need for a safety-critical system. On the other hand, a fault with a cryptographic function could lead to a breach of security or financial loss and hence a security-critical system is required. However, in current times, such a breach in security could typically result in a successful terrorist attack, which could lead to loss of human life, and this leads to the need for a security-critical system to be a safety-critical system.

It can be argued that both types of system require strong partitioning in order to prevent unintended interactions. Safety-critical systems define five levels of failure conditions to which software might contribute: catastrophic, hazardous, major, minor, and no effect. Software functions of varying levels of criticality are often hosted together (on the same module for example) and therefore interference must be considered and prevented.

For security-critical systems the Multiple Independent Level of Security (MILS) specification has been developed. The core of MILS is the separation kernel (SK), which allows multiple software functions from different development and verification sources to share common resources such as CPU, but have no unwanted interference. This security-critical partitioning using formal methods needs enforcing and proving (i.e.: can the SK really do it) and this potentially drives architectural differences.

These different standards and specifications do not necessarily cover the same areas however and this leads to more confusion as the safety-critical standards ensure that the code is well constructed whereas the MILS specifications are not designed for this. Also is it therefore necessary, and indeed, is it possible to interchange these two types of systems and is that sometimes, never or partially? Are the technical differences between the two types of systems too great to support this? Clarity and new understanding is required in this area particularly in the light of the reasons for the Nimrod disaster recently released.

Dr Adele-Louise Carter, Fellow of the BCS, The Chartered Institute for IT and Member of its prestigious Security Forum and Software Manager at REP, brought the need to debate this hot topic to the three organisations. All parties readily accepted the challenge and Adele worked with the organisations to bring all the relevant experts together to consider this area in detail and look at ways of taking it forward. The work has culminated in this report, authored by Adele.

This paper will detail the results, discussions and conclusions that emerged from the resultant Technical Day, Safety-Critical versus Security-Critical Software, on 6th May 2010 at the BCS premises in Covent Garden. Areas covered included policies, processes, staffing, risks and technical aspects, and business and commercial considerations.

2. Research method

Qualitative Research was undertaken using the Action Research methodology utilising focus groups and information exchange sessions. The idea behind Action Research is that the actual activity of undertaking this research contributes to taking forward of the ideas raised within the research. The outcome of the research is changed or influenced literally by its undertaking.

Significant discussion on this topic has been taking place for a number of years and Adele felt that it was time to draw all the previous work undertaken in this area together, collate this knowledge, analyse it and draw conclusions and support these conclusions with an action plan. To this end, Adele pulled on all her work contacts to establish a Technical Day which drew together as many of the most notable leading professional and academic businesses in these two domains to do just this. The day consisted of six key speeches on aspects of safety and security-critical software, a series of round table focus groups to explore specific subjects in more depth and a Q&A (Question and Answer) session to further raise issues and define the next stages. Each focus group was chaired and facilitated by REP volunteer staff.

All the attendees, speakers and focus group contributors at the technical day were carefully selected in order to ensure all angles were represented and therefore significant progress able to be made. Their attendance in such financially tight times shows the commitment of the community to this problem area. Those recommended from initial invitees provided reasoning for their inclusion which actually further contributed to ensuring all areas were covered. Unfortunately, on the actual day, five representatives from the Security-critical arena were absent. Potentially, this limits the number of attendees from a purely security background.

The businesses represented can be divided as follows:

- Large companies in the areas of Engineering, IT and Science including: BAE Systems, QinetiQ, Rolls-Royce, Invensys Rail, Logica, NATS, Atkins, Fujitsu and Ultra Electronics;
- Small and Medium Enterprises (SMEs) such as: Resource Engineering Projects (REP), Trango, BBS IT Associates Ltd, Amethyst Risk Management Ltd and a number of Independents;
- Leading academics from the Universities of York, Newcastle, City, Gloucester, Middlesex and Surrey;
- Government and associated organisations including: CPNI (Centre for the Protection of National Infrastructure), CESG (the National Technical Authority for Information Assurance), TSB (Technology Strategy Board), SITC (Security Innovation and Technology Consortium) and the Electronics KTN (Knowledge Transfer Network);
- Professional bodies: BCS, The Chartered Institute for IT, Safety Critical Systems Club (SCSC), the Royal Academy of Engineering and the IET (Institute of Engineering and Technology).

This representation meant that software experts from all the relevant industries were in attendance, although in different quantities such as: transport; rail, aerospace and automotive, NATS (National Air Traffic Control); MoD, Defence and Homeland Security, the energy and medical sectors.

3. Data presented

The following topics were covered at the Technical Day.

1. The methods and practicalities of combining safety and security assurance – this highlighted the paid work to date in this area.
2. What makes safety critical software different?
3. Risk management and assurance from the security world.
4. Security and Safety or Malicious and Accidental.
5. Considerations when using COTS software within safety-critical systems.
6. Security and safety in networked information systems: issues and challenges.

The cost of safety re-certification relates to the size and complexity of the system and not the actual change, which would be preferable. To such an end modular construction and certification would therefore be more desirable. However, sometimes there is an issue when there are conflicts between safety and security. This could be reduced as there is common ground in terms of risk and threats and assurance methods can usually be identified.

Both areas could do with common terminology, although this may raise more cultural issues rather than practicalities of terminology. The difference with security is that it may have to deal with intelligent malicious agents, not just flawed operators. Currently safety can thus use more statistical methods than security. In the safety area a complete specification of analysis can be undertaken such as in the case of an accident. Within security this is much tougher and the facets can often change, such as 'motivation'.

When it comes to impact, it is much tougher to quantify this for security. In security for example, risk can often be classified, unlike as yet in the safety arena. The traditional view is that malicious issues are not considered in safety. Safety has got away with assumptions of ‘no maliciousness’ and ‘no common mode failure’ for years but can no longer do this.

As yet there is no conclusive proof of software causing a fatality in aviation and it could be that there are lots of standards for the software. In fact the latest standard DO178-C is about to be published and will cover complex and potentially non-deterministic object oriented techniques. It will also replace ‘text’ evidence with ‘formal analysis’ evidence and use modelling tools in development and verification. The latter though is seen by some as being rather contentious.

Risk management seems to be at the crux of the debate. CESG found that much of the equipment that is tested was not robust to electronic attack. Many of the incidents CESG respond to are due to software bugs, reduction in the number of these bugs through good development and testing strategies is an important factor in reducing the risks to information. It is possible to consider security and safety as simply different attributes of dependability. Also within safety software, COTS products need to deal with unexpected inputs and be compatible with the systems they support. This can easily be applied to security.

Finally security can be considered to be about protecting valuable information assets, in general or via computer networks and it can be considered to be about how exploitable a vulnerability might be (and how determined the attacked is); whereas safety is about protecting physical assets (life, property, environ) and this is more complex now and its likelihood of occurrence is often guessed.

4. Results

There was a staggering amount of agreement within the Technical Day community over the issues surrounding safety versus security-critical software. The overwhelming view was that this is a simple issue to resolve in theory, but in practice it is extremely complex with a large cultural hearts and minds exercise needed to be undertaken in order to establish any improvement in the security-critical software.

The attendees consisted of a significant collection of professional and academic experts within this area and to add to this, those leading Governmental bodies and Institutions that might need to be part of any changes required. The overriding point that was raised was that 90% of security-critical software contains bad coding bugs which detracts from making them secure and this was unanimously viewed as of paramount interest by the attendees. It was agreed that that one of the key ideas that emerged from this was the lack of standards in security-critical software, unlike in safety-critical software and that the BSI should be approached to take this forward.

Below are the results from each of the round table focus groups.

4.1 Real life problems of safety and security-critical software

- Standards and guidelines in safety-critical software are assumed to be sufficient, for example DO 178B in use on the Boeing 777 still allowed room for error.
- Further to the above, standards such as DO 178B are software specific and do not necessarily look at the system or environment as a whole.
- Levels of assurance are inconsistent across different industries and indeed different parts of the world and the quality of the software can differ.
- There are too many standards. For example safety has a standard for each industry: aerospace, defence, automotive, rail, nuclear etc.
- Integrating systems such as SCADA, which is dated, with the outside world can be problematic due to the rate of development. Also certain systems are not developed to be integrated or seen by the outside world.
- There is a pressure to create saleable products from software that is perhaps not of the desired quality.
- The long term cost is thought to be high when 'doing things properly'. Similarly development (e.g. producing prototypes) is expensive and not always possible due to the short lead time.
- Supply chain is passing risk/liability further down, this affects the cost and forces the behaviour of sub-contractors by covering all bases when this is not always necessary. Is this enhancing the overall quality or just making it more expensive for the fear of legal action later on?
- One possible solution would be for the regulatory bodies to liaise across software industry, not just safety or security. Alternatively there could be a body for each that work together.

4.2 Technical similarities and differences between safety and security

- The safety community suffers from having too many standards and it would be virtually impossible to create one common standard.
- The security community doesn't tend to share information and this can hinder the learning process.
- The security industry could improve vastly but implement some of the processes that have been used in the safety industry for several years. The security industry also needs to spend a little more time employing good software practices (mainly in terms of verification and validation).
- The security industry reacts to change very quickly. The safety industry needs to develop techniques to help improve the time it takes to make changes to its certified system.
- Safety industry has always worked on the assumption that failures and errors are not malicious. This may not be a valid assumption in the current times. This should be taking into account as part of the risk assessment.
- The security industry would benefit from a formalisation of software security levels. A long term goal would be the safety and security industry sharing a common set of levels.
- The security industry is always going to suffer from not having a configured working environment. Safety generally has a stable known working environment which makes development a lot easier.
- Regardless of standards (both security and safety) the certification authorities need to help developers. Concerns were raised with the certification authorities' statement that the compliance to standards may depend on an individual.

4.3 Combining processes and working practices

- Security-critical development has no real process or standards to aid a consistent approach to software development. This was evident by the largest problem with security-critical software, was not the failings of either architecture or systems requirements, but by the number of errors found in the software.
- Conversely it was believed that the process of developing the software in the safety-critical arena was robust, reducing the likelihood of error propagation in the development but perhaps where it lacked control was at the systems requirement level, specifically in addressing the potential of malicious error injection.
- It was generally agreed that systems should not be treated as safety or security system but as a safe secure system. Bringing the prescribed software development process from the safety standards into the security arena. Additionally bringing security risk hazard analysis principles into the safety arena. Ideally this would lead to a general set of processes which could be applied to either, the question is would the certification assessors and the industries be able to agree on this?

4.4 Skills swapping

- It was unanimously agreed, not just in the discussion but from the speakers, that all systems should be developed as safe secure systems. As a result those skills that are currently unique in each arena should be pushed as high up the software life-cycle as possible to allow for a complementary software skill set in software development (tools and language dependent). This would require a common development process rather than a skill change.
- This would lead to the conclusion that risk and hazard analysis, for both a security and safety assessment, should be conducted and therefore requires skills from both arenas, to define a set of safety and security software requirements to flow down through a standard development process. Independence of this skill set at this high level may be required though to ensure there is no bias towards contradicting risks.

4.5 Safety and Security-Critical Software from a risk perspective

What are the risks / barriers to implementing safety and security systems as one discipline?

- There are many standards in the safety industry, such as, DO-178, 61508, 26262 to name just a few. These standards have a lot of common requirements, but also differences, not least in the definitions and labelling of the SIL levels (SIL A is the highest for Aerospace, SIL 4 is the highest for 61508 and ASIL D is the highest for 26262). With this in mind, trying to standardise the security needs into these standards or into a generic standard would be extremely difficult.
- There are ongoing internet discussions about how the different Safety Integrity Levels of all these standards map onto each other and whether they are equivalent or not. For example there are suggestions that ASIL D in 26262 is equivalent to SIL 3 in 61508 and that there is no equivalent to SIL 4. Therefore, where currently industries can claim that they have developed their software to the highest integrity (for that industry) they might not be able to if there was a common standard. This could open the door for litigation based around the premise that the software could have been developed to a higher safety level. It was felt that for this reason there would be pressure against any combining of standards and industries requirements.
- Discussions amongst the group concluded that while the techniques and methodologies used were similar, and could be considered as transferable, the terminology between the different disciplines was different and therefore placed conceptual barriers to this transferral. For instance, if the safety / hazard analysis from the safety domain and the security analysis from the security domain were both considered as risk analysis then the use of techniques could be more easily standardised and the two disciplines seen as one.
- Questions of certification and accreditation were seen as a barrier. Would the applicant need to have their system qualified in both domains along with its associated costs. But more fundamentally, would the applicant need to develop the software with two sets of plans, development paths and results, just to comply with the requirements of each discipline.

4.6 What is needed to change standards?

The topic was broken down into four questions:

1. Who should do it?
2. What does it need to say?
3. How do we get it adopted?
4. Do we need to revise existing standards or propose a new one?

- It was agreed that a cross-industry approach was best, to avoid the situation that now exists in the safety critical world, where each industry (aeronautical, railway, automotive etc:) has its own standard.
- Whether a combined safety and security standard would be feasible was considered. It was felt that in the area of risk management, at least, it was possible. Given the results, the approach recommended was for an overarching document that would refer to the 'best parts' of the various existing standards.
- An initial white paper was proposed, offering guidance and inviting feedback. This would lead to an initially UK standard (BSI) and then ultimately, a European one.
- It was felt that changing existing standards was unrealistic. In particular, recent experience in revising safety standards in the railway industry indicated that such a process would be much too slow.

4.7 Next steps

- Adele to compile report for IET to stimulate further discussion.
- Identify key stakeholders from relevant organisations and approach with relevant concepts (e.g. Government, CESG, Cyber security).
- Use TSB Secure Software Development Partnership as a vehicle to provide further discussions or form own working party with key industry figures, covering both safety and security domains (e.g. BAE Systems, REP, Qinetiq, Logica).
- Could REP consider security further as part of their normal working practices and build this into their quality assurance processes?
- Use working party to assess current difficulties within the security industry, for example the "need to know" culture that has been developed (CESG could be used for this) and also for assisting with developments within academia (e.g. City University).

The Q&A session covered standards, architecture, requirements and users of the software. It was concluded that there was a need for a common integrated approach and common goal between safety and security-critical software.

5. Conclusions

The overall conclusions from the event were:

1. A core community consisting of all the relevant parties is required in order to keep everything together. The attendees at the Technical Day were considered to represent these and perhaps join forces with the TSB's Secure Software Development Forum.
2. Both domains have common issues. Both have to deal with the implications of risk and the need to minimise those risks. One has a security risk assessment and the other a safety case.
3. The lack of common terminology between the two areas produces an artificial barrier when you consider that the safety and security-critical domains are as one.
4. The techniques for determining safety requirements and security requirements are fundamentally the same.
5. Security software tends to have a rapid development life cycle to mitigate threats quickly, which tends to result in a level of software errors.
6. Safety software has a longer development life cycle and hence is less prone to software errors?
7. Both domains are suspect to system errors i.e.: failure to define correctly what you required the system/software to do.
8. Education in addition to regulation and standards was required in security-critical software and this should potentially stem from Universities. The Secure Software Development Partnership of the TSB wrote a white paper on this, Software Security Failures – Who should correct them and how?, Issue V 1.0 June 2008, Bill Whyte and John Harrison.

6. Recommendations

The Technical Day attendees highlighted the following main recommendations:

1. The security domain would benefit from using software development techniques from the safety domain but cost and time-scale implications may be a barrier. There is regulation in the safety critical domain but it is not apparent for software developed for the security domain and therefore this area should be investigated.
2. The safety domain would benefit from considering malicious attack on its systems and defining requirements and techniques that mitigate these vulnerabilities.

3. Safety and security-critical systems should be considered as one, safe secure systems and a common terminology would go a long way to supporting this.

7. Action Plan

The following action plan has been formed to ensure that these ideas are taken forward and do not just end up written into another report that is forwarded around and ignored.

1. Identify further key stakeholders (if necessary) from other relevant organisations that have influence and power and approach as appropriate.
2. Further Meetings with CESG, CPNI, Office of Cyber Security, OGC Buying Solutions at Norwich, H&S Executive and the Nuclear Executive to discuss measures to take matters forward.
3. Potential TSB Submissions to their current competition on trusted systems.
4. Web report to be provided for awareness on CPNI/CESG websites.
5. Paper to be produced for IET Conference.
6. Support TSB's Secure Software Development Partnership as a vehicle to provide further discussions covering standards, tools and techniques, education and training, embedded real time and formal methods.
7. Contact BSI about security software standards as we have the group for creating these standards via the CPNI.
8. Ensure that the community from the technical day is maintained and possibly use it to assess current difficulties within the security industry, for example the 'need to know' culture that has been developed and also for assisting with developments in academia.

Finally, as all work to date to create this report has been undertaken on a voluntary level, in order to take anything further, it is now necessary to consider funding opportunities. In addition, a suitable 'Character' is required to take these actions forward. To date some work on these actions had already been stimulated by Dr Adele-Louise Carter, particularly with TSB and CPNI.