Journal of Information Security, 2020, 11, 292-303
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

Concepts of Safety Critical Systems Unification Approach & Security Assurance Process

Faisal Nabi[1*], Jianming Yong[1], Xiaohui Tao[1], Muhammad Saqib Malhi[2], Umar Mahmood[2], Usman Iqbal[2]

[1]School of Management and Enterprise, University of Southern Queensland, Toowoomba, Australia
[2]Melbourne Institute of Technology, Melbourne, Australia
Email: *faisal.nabi@yahoo.com

How to cite this paper: Nabi, F., Yong, J., Tao, X., Malhi, M.S., Mahmood, U. and Iqbal, U. (2020) Concepts of Safety Critical Systems Unification Approach & Security Assurance Process. *Journal of Information Security*, **11**, 292-303.
https://doi.org/10.4236/jis.2020.114018

Received: September 23, 2020
Accepted: October 24, 2020
Published: October 27, 2020

Abstract

The security assurance of computer-based systems that rely on safety and security assurance, such as consistency, durability, efficiency and accessibility, require or need resources. This targets the System-of-Systems (SoS) problems with the exception of difficulties and concerns that apply similarly to subsystem interactions on a single system and system-as-component interactions on a large information system. This research addresses security and information assurance for safety-critical systems, where security and safety are addressed before going to actual implementation/development phase for component-based systems. For this purpose, require a conceptual idea or strategy that deals with the application logic security assurance issues. This may explore the vulnerability in single component or a reuse of specification in existing logic in component-based system. Keeping in view this situation, we have defined seven concepts of security assurance and security assurance design strategy for safety-critical systems.

Keywords

System Security, Assurance of Component Function, Safety-Critical Software, Software Assurance

1. Introduction

The integration of components into industrial control systems such as railway control and management systems (CCS) is ongoing of commercial off-the-shelf hardware and software (COTS). However, the use of COTS components in a pre-owned security framework results in new security risks. The interplay of security is an important field of study in which several questions still need to be addressed. To mitigate risk and ensure the programme is dependable and secure;

security assurance is an essential part of the safety-critical software development process. Deficiencies in infrastructure and deficiencies also can lead to software bugs and abuse by hackers and offenders seeking to manipulate flaws in the tech industry. Testing, accreditation and evaluation are carried out to justify the level of assurance of safety of logical function during the intercommunication interaction process. This strategy is applied at design stage that refers to traditional use to increase the trust of the programme in the programme validation process [1].

Software assurance during the engineering/development process has been an integral aspect of contemporary safety-critical systems' overall innovation, ranging from weapons, avionic, even automotive control systems, industrial control systems and medical equipment. Software is used for tracking and regulating physical processes in these systems increasing failure may lead to loss of life or other catastrophic malfunction. Therefore, software assurance for safety-critical systems performs a role as backbone in commercial-off-the-shelf component-based system [2].

Ever more software, including embedded systems, is no longer purpose-built in security systems. Instead, they are used (or reused) for COTS, GOTS Government off-the-shelf for software and hardware, open source, and other non-developmental applications, often without alteration or advanced setup changes. Much of this no developmental software—especially COTS and open source software—is component: stand-alone software pieces which can be used as a building block for creating larger and more complicated systems of software. The smallest independent decomposition unit in a software-based system may or may not be a component [3]. In certain cases, components with smaller modules are assembled. To be usable as a component of a broader framework, an autonomous programme must provide interface(s), typically standardising to allow the integrating or mounting of other components. In this case, degree of component assurance and system safety is foremost priority in information assurance for safety-critical component-based software systems in organizational [4].

The most important aspect in security assurance of computer-based systems is inter-component specification. Interactions between the systems may be separated by one component and another function consumption [5]. The service (function or calculation) provided to another component can be specified as a contract between the consumer component and the supplier component by one component and the services requested from the other component and details of interface(s) by which these provisions and applications are made [6]. The expectations one component has regarding the contractual commitments other components may meet are clearly specified as the preconditions or constraints the component sets on the other components with which it may communicate.

In this research, we are going to address the effects of security and information assurance on safety-critical component-based software systems, which discusses the security and safety in implementation & development process of systems. This requires a strategy to deal with the application logic security concerns

that may explore the vulnerability in single component or a reuse of specification of existing logic in component based system.

2. Research Method

We have used the applied research method for our research work. The method has a subclass called research evaluation. In this method, we address and assessment analysis is a kind of analytical study evaluating current research knowledge that is subject to empirical study results or to informed decision-making [7], for example, a scientific method of investigation because it applies existing scientific knowledge to consolidate situations to perform appraisal analysis to decide the research problem and proposed theory. Therefore keeping in view this research method, we have proposed seven concepts for information assurance for safety-critical component-based software.

3. Background of Research

The speeding up of attacks as well as the obvious shift to further vulnerability appear to mean that our ability to resolve attacks diminishes and the divide between attacks and information defence broaden. Most of the modern information security is based upon concepts as defined by Saltzer and Schroeder in the 1974 ACM Communications article entitled "The Security of Information in Computer Systems". Protection was characterised as "techniques to monitor who may access or change the device or information stored therein" and the three key categories of concern were described: secrecy, credibility and availability [8].

We constitute the security assurance in cyber security and information assurance for safety-critical component-based software systems as:

"*Software Assurance considered as security is a trust degree of protection from software several bugs, designed purposefully or unintentionally, are implemented in the software at any point during its life cycle such therefore the software works as intended*".

With vulnerability breaches expanding through ransomware, bugs, and injections of structured query language (SQL), cross-site scripting, etc., these challenges have altered the structure and functionality of the programme. It has proven to be incredibly inadequate to rely solely on identity security. In addition, the importance of software in networks has evolved such that software now manages the majority of functionality and increases the effect of security failure [9].

The convergence and interoperation of security and safety-critical systems is becoming more and more apparent. It makes sense, therefore, to create an overall concept of software assurance covering safety and protection. The various methods proposed by the current concepts emerge in several cases from threats associated with complex structures [10].

Furthermore, the acceptance of commercial off-shelf (COTS) and open-source

software as modules within a framework creates additional challenges for successful operating protection. The resulting operating systems combine applications from a wide variety of sources and assemble each piece in a distinct manner [11].

Systems cannot be built to eliminate safety risks but have the ability to recognise, resist and recover from attacks. The system should be prepared for implementation and maintenance in the initial acquisition and design. In order to ensure successful organisational protection over time, assurance must be scheduled over the life cycle [12].

Now we use the following concept of component-based software lifecycle assurance built for:

Technologies and procedures are implemented to obtain the required degree of trust that applications and services work as expected are free of unintentional or deliberate flaws and have threat-friendly protection functionality as well as recovery from intrusions and failures.

4. Existing Research Review

There is not very much research work is done in the domain of security assurance unification process of safety-critical component-based software systems. However, we have considered some important work to cite the research work to underpin with the effort of research design.

According to Faisal Nabi 2017, proposed security assurance unification process that defines.

Author describes the architecture in two stages of abstraction of an information system.

1) The design level of the method explains form for architectural form levels to be implemented at the highest-level abstraction.

2) The architecture definition of a logical part.

To ensure safe deployment, protection needs to be applied using a design approach, rather than implementing a layer in the framework, by co-operation with the above-mentioned core elements of the security assurance process. The architecture can therefore be extracted by means of protection the assurance protocol course [1].

According to Tim Kelly 2019 explained that an alternative solution by establishing a structure for compliance and data assurance (SSAF) focused on the fundamental set of standards of security. Instead of a popular co-assurance, which has identified major disadvantages, protection and safety should be individually co-assured. This often permits different processes and practitioners' skills in each area. With this arrangement, attention is transferred from simpler convergence to integration through the correct knowledge exchange with the synchronisation activities at the right time [3].

According to Marsha Chechik (B, Rick Salay, Torin Viger, Sahar Kokaly, and Mona Rahimi 2019), Addresses the Test cases, test data, human decision or a

mixture of these will provide data for software assurance. This means that experts strive to construct (safety-critical) structures with caution and to express that reasoning according to well-founded methodology in a safety case that is eventually tested by an individual. However, tech has deeper origins in uncertainty, the most complicated open world features (for example, a self-driven vehicle's understanding of the state of the earth) often are not entirely predictable or not cost-effective; computing applications are also put in dangerous conditions, and there can be inconsistencies [2].

Impact of Safety & Security Risk

Risk factors are security threats that add safety hazards to a system when discovered. Security impact risks are directly risks to the system's Integrity and availability characteristics.

Integrity is a security feature directly connected to durability and trustworthiness: it depends on its durability that it is not modified inadvertently, by mistake, by an unwanted entity, or through illegal means, either accidentally or purposely. The trust of the device is not compromised such that bugs or deceptive reasoning can be implemented [13]. The credibility of the system is therefore critical because unauthorised and unintended changes can only impair the system's ability to run effectively, but may also prone the system to unnecessary compromises and/or incorporate unauthorised functions. In all cases, such modifications often wrongly indicate any of the conclusions based on careful examination and review of the system before it is implemented.

Assuring the consistency of the system guarantees that the system is intact and that the system decisions are valid.

Another security feature is closely associated with system reliability. To be available the system must, as defined in its specifications (e.g. 99 percent of the time, 95 percent of the time, etc.) be active and open to its intended users. The availability is similar to the "required uptime" and "quality of service", only that it not only covers the system's operating consistency and the consistency of system connectivity for those who use it.

5. Proposed Concept of Security Assurance Safety-Critical Systems

Components are engineered primarily to be combined into systems, and they really require security eventually. Composing security cations into broader systems is not only a non-trivial task, but also one of the toughly unanswered information security issues, in order to deal with this issue in the business logic of a compound (logical component-ware interface-orientated design) in an e-commerce application. For logical component-based fast advances, and for an increasingly increasing business process logic in e-commerce systems, we need the convergence of security process resources as shown in **Figure 1**.

In order to reach a desired degree of trust for software security assurance, we recommend the seven concepts aimed at solving the problems associated with

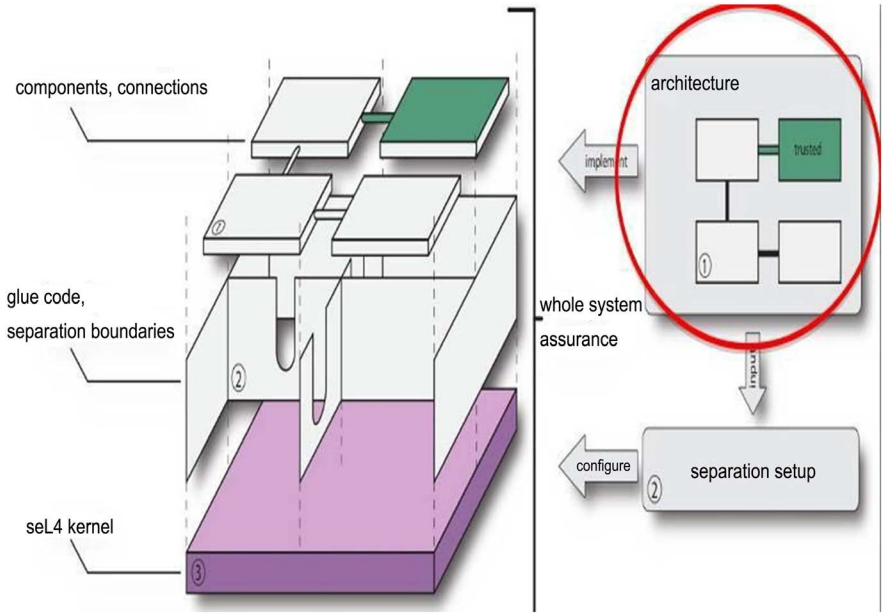

Figure 1. Security assurance process and properties of unification.

the information assurance for safety-critical component-based software systems, construction, deployment and retaining of systems.

1) Risk guides decision-making in assurance. A risk-taking perception guides decision-making. Organizations that do not obtain reliable security assurances experience danger from efficient attacks on infrastructure and systems. They can use assurance options as a function of their perceptions of a threat of similar attack and the anticipated effect, such as strategies, procedures, methods and limitations, if this threat is understood. Organisation, because they struggle to grasp their challenges and impacts, may falsely interpret risks. Efficient security allows businesses to share risk awareness with both partners and participants of the project.

2) Risk issues shall be associated with both stakeholders and strategic aspects intertwined. Highly linked networks such as the Internet require coordination of risk between all players involved and all technical elements linked to them; otherwise, at various points in the relationships, important risks are overlooked or ignored. When all are deeply intertwined, it is not enough to consider just deeply essential elements. Interactions are carried out at different levels of technology (e.g., network, security, infrastructure, and applications) and are assisted by a number of functions. Security at any of the stages may be applied and, if not well planned, may clash. Effective assurance requires clear identification, response to risk at all levels, and positions related to interactions.

3) When proved trustworthy, dependencies are not to be trusted. Because of the extensive use of digital supply chains, the guarantee of an automated commodity relies on the judgments of those in terms of commitment and the degree of faith imposed in them. All the guarantees' shortcomings of each communicating component come from the optimised applications. In addition, any oper-

ating function, including utilities, security software and other programmes, is subject to the guarantee of any other function unless unique constraints and controls are in effect. A company still relies on the guarantee decisions of others. There is a chance. Organizations have to determine, however, how much confidence they put in their reliance on a practical appraisal of risks, consequences and opportunities across diverse experiences. Dependences are not stagnant and businesses have to revisit confidence ties on a daily basis to assess adjustments to be rethought.

The following examples define assurance damages from weakness:

- Centralized technology vulnerabilities (e.g. operating systems, programming environments, firewalls, and routers) can act as publicly accessible software vulnerability entry points.
- The use of several common technology construction development tools efficiently assures the resulting digital product. The tool manufacturers may introduce vulnerabilities into software products.

4) Attacks are expected. The secrecy, credibility and availability of technical resources are sacrificed for a wide group of assailants with increasing technological capabilities. No security from attacks is flawless and the profile of the attacker continues to evolve. In order to reach a consensus (known as social-technical response), attackers are using technologies, procedures, norms and practise. Some threats are using technologies, and others create unique conditions to exploit protections. They are the way we use technologies.

5) Ensuring that the software assurance concerned needs good teamwork. Organizations must extend security throughout their employees, procedures and technologies while assailants seek all potential access points. In addition, organisations must specifically define at an adequate level the policy authority and obligation for ensuring that corporate participants engage efficiently in cyber security. This theory presupposes that everybody is confident, but generally, it is not. Therefore, organisations need to prepare staff to maintain tech.

6) The guarantee is creative and well planned. Assurance may have a bridge between software and network administration, design and service and is extremely susceptible to improvements in any of these fields. To preserve this equilibrium, it is important to respond to frequent shifts, interconnections, organisational use and risks of applications. This is not a one-time occurrence, because transition is regular. It needs to proceed by organisational monitoring after the initial organisational deployment. This must be incorporated into the appropriate promise that companies require. This will not be added later. Every time, nobody has money to overhaul structures.

7) An overall assurance assessment and evaluation process should be implemented. Organizations cannot cope with something they cannot calculate, and consumers and consumers of technology will not take responsibility for policies until they take responsibility for it. If outcomes are tracked and calculated, Confidence cannot compete with other competitive needs effectively. To determine

organisational assurance, all socio-technical elements like policies, processes and procedures need to be connected together. More efficient assurance process responds and rebound more quickly. They will benefit about their and others' reactive reactions, and predict and identify threats more carefully.

For Example: Code faults is a standard implementation metric, and can be useful for code consistency, but is not acceptable proof for general certainty since it gives little insight into how code functions in an operating environment. Concentrating and systematic steps must be taken by organisations, to ensure sound protection is established for the components and efficient assurance of the relationship within components.

Evaluation Security Assurance Level Analysis Chart (Table 1)

The security assurance is achieved through the validity of empirical analysis of proposed concepts and the process of system assurance, as it is explained in the given below model. This depicts the seven stages of security assurance level for information assurance that is concluded based on proposed seven concepts for safety-critical component-based software systems.

Table 1. Evaluation of security assurance in safety-critical systems.

Evaluation assurance level	What is tested	Description
1	Functionality	Evaluation provides independent testing against a specification and an examination of the guidance documentation. Used when confidence in correct operation is required but the threats to security are not viewed as serious.
2	Structure	Evaluation provides a low to moderate level of independently assured security as Required by vendors or users.
3	Methodology	Evaluation provides an analysis supported by testing, selective independent confirmation of the vendor test results, and evidence of a vendor search for obvious vulnerabilities.
4	Methodology and Design	Evaluation provides a moderate to high level of independently assured security in conventional commodity products. Testing is supported by an independent search for obvious vulnerabilities.
5	Semiformal Design	Evaluation provides a high level of independently assured security in a planned development, with a rigorous development approach. The search for vulnerabilities must ensure resistance to penetration attackers with a moderate attack potential.
6	Semiformal Verified Design	Used for the development of specialized security products, for application in high risk situations. The independent search for vulnerabilities must ensure resistance to penetration attackers with a high attack potential.
7	Formal Design	Used in the development of security products for application in extremely high risk situations. Evidence of vendor testing and complete independent confirmation of vendor test results are.

6. Designed Defensive Strategy as a Solution to Deal Business Logic Layer Concerns

This part of strategy will provide a strong risk management control plan focusing on providing rigours component ware assurance for rapid development of CBSD business application logic for safety-critical component-based software systems and its applications in e-commerce domain.

Key elements of problem solution follow: 1) Strong risk management plan; 2) Solution artefacts; 3) Security characteristics of component-ware components.

1) Strong risk management plan:

Ensure that every aspect of the application's design must be clearly & sufficiently detailed to understand every assumption and designed function logic within the application by designer.

Mandate that all CBSD should be clearly commented to include the following information throughout.

a) The purpose and intended use of each component (if component code available information of code, if not, its functional business logic within the component through usage contract description).

b) The assumptions & logic made by each component about anything that is outside of its direct control.

c) Reference to all client-component which makes use of the component clear documentation to this effect could have prevented the logic flaw within the online registration functionality.

(Note: Client here dose not refer to the user-end of the client-server relationship but to other component (code) for which the component being considered is an Immediate dependency.)

2) Solution Artifacts: As that there is no unique signature by which logic flaws in component-Based-Rapid developed web software application can be identified, because there is no silver bullet so far developed which could protect.

Good Practice: Good practice that can be applied to significantly reduce the risk of logical flaws appearing within component-based-development and its logic.

3) Security Characteristics of Component Ware Components: Since a software component can be regarded as an IT product or system, it is natural to use the Common Criteria in assessing its security properties. The Common Criteria provide a framework for evaluating IT systems, and enumerate the specific security requirements for such systems. The security requirements are divided into two categories:

- Security functional requirements
- Security assurance requirements

The security functional requirements:

Describe the desired security behaviour or functions expected of an IT system to counter threats in the system's operating environment. These requirements are classified according to the security issues they address, and with varied levels of security strength. They include requirements in the following classes: security audit, communication, cryptographic support, user data protection, identification and authentication, security management, privacy, protection of system security functions (security meta-data), resource utilization, system access, and trusted path/channels.

The security assurance requirements:

The security functional requirements mainly concern the development and

operational process of the IT system, with the view that a more defined and rigorous process delivers higher confidence in the system's security behaviour and operation. These requirements are classified according to the process issues they address, and with varied levels of security strength. The process issues include life cycle support, configuration management, development, tests, vulnerability assessment, guidance documents, delivery and operation, and assurance maintenance.

Figure 2 presents the idea of security assurance process based on layer of security assurance of component-based software application logic for e-commerce systems. This process is also helpful for developers of safety-critical component-based software systems, while reusing specification of existing logic for the current system.

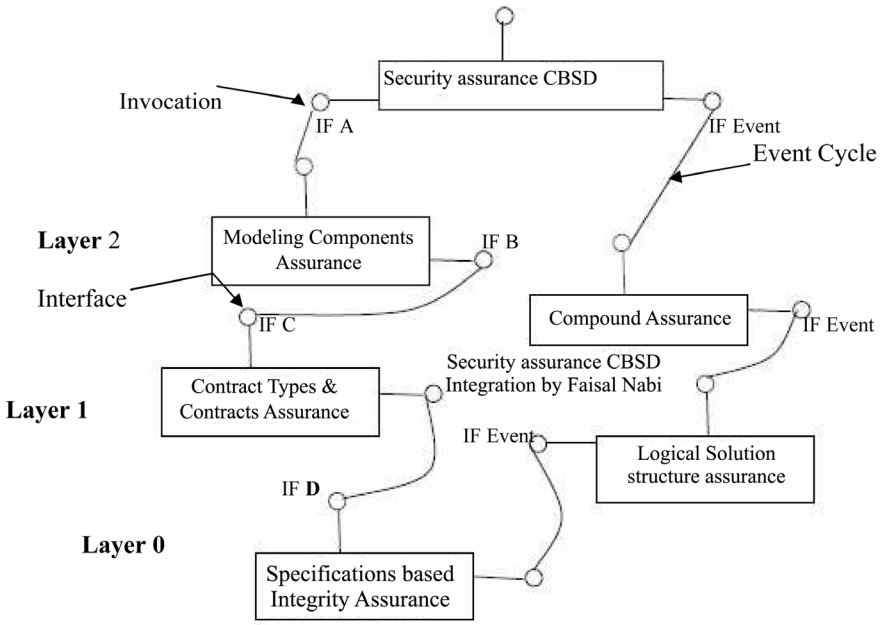

Figure 2. Design strategy process for security assurance business application logic.

Therefore, it is important that safety-critical systems, those are almost in daily use of human interaction and from simple system to complex systems of component based require assurance before passing through development phase that guarantee the safety of the system in various environment.

7. Conclusion

This paper addressed some of the key problems and information gaps in defence and protection in big, complex systems. These flaws are due to gaps between protection and safety systems, how threats are portrayed and clarified, and how claims should be viewed as templates. The seven concepts were described as a Safety Assurance and design security assurance strategy mechanism solution for the independent system or component to the difficulties of developing a mechanism that synchronises separate security and safety assurances and provides a

more sophisticated and complex form of evaluating impacts. The seven concepts are blue print capable of modifying the relationship of the intelligence and security sectors and security design strategy process of modelling help developers to make sure the system security assurance at SDLC stage, which is proved to be as a guideline for safety-critical component-based software systems.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Nabi, F. and Nabi, M.M. (2017) A Process of Security Assurance Properties Unification for Application Logic. *International Journal of Electronics and Information Engineering*, **6**, 40-48.

[2] Chechik, M., Salay, R., Viger, T., Kokaly, S. and Rahimi, M. (2019) Software Assurance in an Uncertain World. In: Hähnle, R. and van der Aalst, W., Eds., *FASE* 2019, LNCS 11424, 3-21. https://doi.org/10.1007/978-3-030-16722-6_1

[3] Kelly, T. (2019) An Assurance Framework for Independent Co-Assurance of Safety and Security. New York University Press, New York.

[4] Czarnecki, K. and Salay, R. (2018) Towards a Framework to Manage Perceptual Uncertainty for Safe Automated Driving. In: Gallina, B., Skavhaug, A., Schoitsch, E. and Bitsch, F., Eds., *SAFECOMP* 2018, LNCS, Vol. 11094, Springer, Cham, 439-445. https://doi.org/10.1007/978-3-319-99229-7_37

[5] Cˆarlan, C., Gallina, B., Kacianka, S. and Breu, R. (2017) Arguing on Software-Level Verification Techniques Appropriateness. In: Tonetta, S., Schoitsch, E. and Bitsch, F., Eds., *SAFECOMP* 2017, LNCS, Vol. 10488, Springer, Cham, 39-54. https://doi.org/10.1007/978-3-319-66266-4_3

[6] Cârlan, C., Ratiu, D. and Schätz, B. (2016) On Using Results of Code-Level Bounded Model Checking in Assurance Cases. In: Skavhaug, A., Guiochet, J., Schoitsch, E. and Bitsch, F., Eds., *SAFECOMP* 2016, LNCS, Vol. 9923, Springer, Cham, 30-42. https://doi.org/10.1007/978-3-319-45480-1_3

[7] Kriaa, S., Pietre-Cambacedes, L., Bouissou, M. and Halgand, Y. (2015) A Survey of Approaches Combining Safety and Security for Industrial Control Systems. *Reliability Engineering & System Safety*, **139**, 156-178. https://doi.org/10.1016/j.ress.2015.02.008

[8] Symantec (2018, March) 2018 Security Threat Report. ISTR Internet Security Threat Report, Vol. 23.

[9] Bird, J. (2017, October) 2017 State of Application Security: Balancing Speed and Risk.

[10] Ullrich, J. (2016, April) 2016 State of Application Security: Skills, Configurations and Components. SANS Institute Survey.

[11] Zakaszewska, A. (2016) Proportionality Approach Model for the Application of ASEMS. BMT Isis Limited (2016, March) (Issue 1).

[12] Finnegan, A. and McCaffery, F. (2014) Towards an International Security Case Framework for Networked Medical Devices. *International Conference on Computer Safety, Reliability, and Security*, September 2014, Springer, Cham, 197-209. https://doi.org/10.1007/978-3-319-24255-2_15

[13] Gehr, T., Milman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S. and Vechev, M. (2018) AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation. 2018 *IEEE Symposium on Security and Privacy* (*SP*), San Francisco, 20-24 May 2018. https://doi.org/10.1109/SP.2018.00058