Postprint of: Górski J., Wardziński A. (2019) Supporting Cybersecurity Compliance Assessment of Industrial Automation and Control System Components. In: Flammini F. (eds) Resilience of Cyber-Physical Systems. Advanced Sciences and Technologies for Security Applications. Springer, Cham. https://doi.org/10.1007/978-3-319-95597-1_4

Supporting Cybersecurity Compliance Assessment of Industrial Automation and Control System Components

Janusz Górski[1], **Andrzej Wardziński**[1,2]

[1)] Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Poland
janusz.gorski@pg.edu.pl, andrzej.wardzinski@pg.edu.pl
[2)]Argevide sp. z o.o., Gdańsk, Poland

Abstract. The chapter presents a case study demonstrating how security requirements of an Industrial Automation and Control System (IACS) component can be represented in a form of Protection Profile that is based on IEC 62443 standards and how compliance assessment of such component can be supported by explicitly representing a conformity argument in a form based on the OMG SACM metamodel. It is also demonstrated how an advanced argument assessment mechanism based on Dempster-Shafer belief function theory can be used to support assessors while analyzing and assessing the conformity argument related to an IACS component. These demonstrations use a NOR-STA tool for representing, managing and assessment of evidence-based arguments, which have been developed in our research group.

Keywords. cybersecurity; IACS component; protection profile; security standards; evidence-based argument; conformance case; certification; tools

1 Introduction

Cybersecurity assessment of an IACS (Industrial Automation and Control System) component involves identification and examination of its critical assets, related threats and security functions which aim at preventing the threats from occurrence and/or from violating security of the assets [1]. For each security function, a set of more detailed security requirements can be specified down to the level where the satisfaction of each requirement can be demonstrated by the available evidence. Specification of critical assets, related threats, security functions and the corresponding security requirements together with the contextual infor-

mation form what is called a *protection profile.* Protection profile is an implementation-independent set of generic security requirements for a family of components and is usually used as the reference in the security assessment and certification process. It is expected that the manufacturer of a component provides evidence demonstrating that the security requirements specified in the protection profile are met by the component. The assessment of the support given by this evidence to the security requirements is part of the component security certification process.

The evidence comes from different sources, including compliance examination based on the submitted documentation, evidence resulting from security testing, and evidence related to development, shipping, installation and maintenance processes of the component. Fig. 1. illustrates how evidence is used in relation to the main elements of the protection profile of a given component.

Fig. 1. Evidence-based cybersecurity assessment schema of an IACS component

In this chapter we present the idea that security of a component can be claimed by building an evidence-based argument which argues that the security functions identified in the related protection profile are adequately implemented by the corresponding security requirements, where the satisfaction of security requirements is demonstrated by the available evidence. In general, such an argument represents a *security assurance case* of the considered component (for a meta-model of assurance cases see [2] and for recommendations related to assurance cases see [3]). Different tools are available to support development of assurance cases. In our

case study we have used NOR-STA [4] which supports integrated management of argument, evidence and assessment.

Following [1, 5] we assume that a component delivered by its vendor is being evaluated against security requirements which are represented in a protection profile specific for a given family of components. We assume that a mechanism for defining, endorsing and maintaining the protection profiles of IACS components is available to the vendors, users and certification bodies. The following are examples of IACS component families [6]: engineering software, firewall, historian station, manufacturing execution system server, Programmable Logic Controller (PLC), Remote Telecontrol Unit (RTU), SCADA client, SCADA server, switch, VPN gateway, WIFI access point.

In this chapter we first introduce the IEC 62443 concepts to which we refer while describing protection profiles of IACS components and then we introduce our case study – a protection profile of the RTU (Remote Terminal Unit) family of components. Then we present how security requirements of the protection profile were represented in the form of an evidence-based argument pattern (called *conformance template*) and how such conformance template could be used to develop a complete conformance argument of a component belonging to the RTU family. This is followed by a demonstration how the argumentation assessment mechanism based on Dempster-Shafer belief functions theory can be used to support compliance assessment of IACS components against the security requirements. At the end, we summarize our experiences in the conclusions.

We demonstrate our ideas using the NOR-STA system for developing, maintaining and assessing evidence-based arguments [7].

2 Related works

Protection Profile is one of the core concepts in Common Criteria [5] and together with the concept of Security Target (ST) refers to the security requirements related to the target object subjected to security assessment. These concepts were used (with some modifications) in [1] and we follow [1] in this respect.

Using evidence-based arguments to demonstrate conformity has been argued in [8] and applied in different domains, including medical, oil and gas, automotive and others. Several researchers attempted to demonstrate and assess security by developing explicit assurance cases. [9] proposes an argument structure that decomposes the main security claim into four sub-claims: C1) *System security requirements are effectively formulated*, C2) *System security requirements are captured in design*, C3) *System implementation is secure*, and C4) *Operational security requirements compliance measures are clearly defined* (effectively put in place). Here, claim C1 corresponds to a Protection Profile and its argumentation strategy is to identify all threats, possible attack surfaces, attack scenarios and effective counter-measures which are translated into implementation and operation related security requirements.

[10] presents an approach where the goal is to demonstrate a set of security capabilities (like Automatic Logoff, Transmission Confidentiality or Cyber Security Product Upgrades) and an argumentation pattern is used to demonstrate each of these capabilities. This approach has been used in IEC 80001 series of standards (in particular part 2-9 published in 2017 includes guidance for use of security assurance cases to demonstrate device security).

A systematic approach to develop an evidence-based argument demonstrating that security requirements are met has been proposed in [11]. The approach is based on incremental development of the security argument as the design and implementation decisions are made and providing evidence in the development and testing process.

3 Introduction to IEC 62443

IEC 62443 is a series of standards and technical reports addressing security assurance of Industrial Automation and Control Systems. The standards apply to manufacturers, integrators as well as to end-users (the standards were initially developed by the International Society for Automation and they are also referred to as ISA99 standards [12]). IEC 62443 consists of several standards covering four areas: general definitions and metrics, policies and procedures for the plant owners and suppliers, security requirements for systems, and security requirements for components.

In IEC 62443, security requirements for IACS components are decomposed into seven *Foundational Requirements* (FR). These FRs are the categories used to organize technical security controls and form the basis for subsequent more specific requirements. They are as follows [13]:

- FR1: *Identification and authentication control* (IAC): necessary capabilities to reliably identify and authenticate all users (humans, software processes and devices) attempting to access the *Target of Evaluation* (ToE) shall be provided.
- FR2: *Use control* (UC): necessary capabilities to enforce the assigned privileges of an authenticated user (human, software process or device) to perform the requested action on the system or assets and monitor the use of these privileges shall be provided.
- FR3: *System integrity* (SI): necessary capabilities to ensure the integrity of the ToE to prevent unauthorized manipulation shall be provided.
- FR4: *Data confidentiality* (DC): necessary capabilities to ensure the confidentiality of information on communication channels and in data repositories to prevent unauthorized disclosure shall be provided.
- FR5: *Restricted data flow* (RDF): necessary capabilities to segment the control system via zones and conduits (communications channels) to limit the unnecessary flow of data shall be provided.

FR6: *Timely response to events* (TRE): necessary capabilities to respond to security violations by notifying the proper authority, reporting needed evidence of the violation and taking timely corrective actions when incidents are discovered shall be provided.

FR7: *Resource availability* (RA): necessary capabilities to ensure the availability of the control system against the degradation or denial of essential services shall be provided.

For each Foundational Requirement, part 62443-4-2 provides a lists of *Component Requirements* (CR). For instance, the following CRs correspond to FR4 (for the full inventory of CRs see [14]):

CR4.1: Information confidentiality – components need to provide for protection of the confidentiality of information in transit.

CR4.2: Information persistence - components need to provide the capability to erase all information, for which explicit read authorization is supported, from components to be released from active service and/or decommissioned.

CR4.3: Use of cryptography - if cryptography is required, the component needs to use cryptographic security mechanisms according to internationally recognized and proven security practices and recommendations.

4 Case Study: Remote Terminal Unit

In this section we present an excerpt from the Protection Profile of a sample family of IACS components, namely the Remote Terminal Unit (RTU) family. This case study has been elaborated by the National Exercise Team in Poland (NET-PL) while working on the validation of the European IACS components Cybersecurity Certification Framework (ICCF) [1].

4.1 Component description

Remote Terminal Unit (RTU), in the following text also referred to as Target of Evaluation (ToE), monitors and controls instruments of SCADA systems used in industrial critical infrastructure processes, like oil and gas pipelines, electric power generation and transmission, chemical manufacturing, physical and technical protection systems, water treatment or others.

RTU main functions include:

- collecting measurements from sensors,
- execution of logic and control calculations,
- user program execution,
- issuing control commands that modify a process,
- communicating with external applications and other devices,

- administration functions to configure or program other functionalities; several administration interfaces are possible: administration console, programming workstation, web-clients,
- supporting removable devices (USB drives, SD memory cards etc.),
- local logging (in particular logging security and administration events),
- remote logging (in particular logging security and administration events).

The usage context of the ToE is presented in Fig. 2. The four parts labelled P1, P2, P3 and P4 shown in Fig. 2 are the interfaces through which RTU interacts with its environment. These parts represent flows (data, control) between RTU and the environment and are included in the scope of security assessment of RTU.

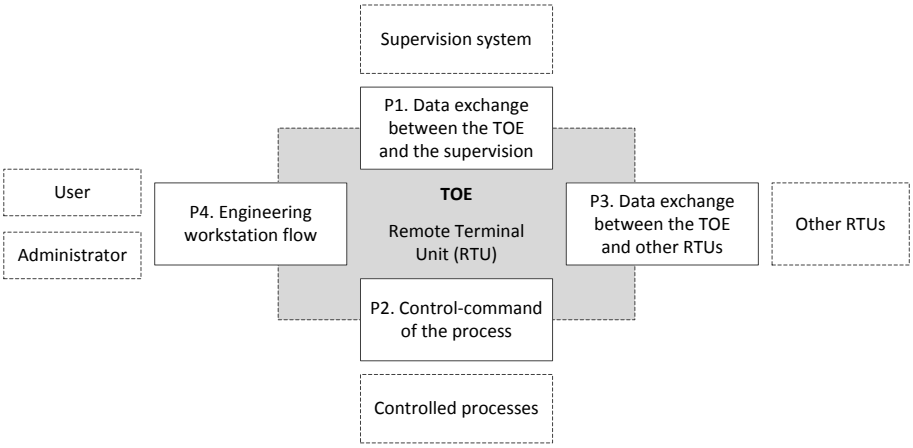

Fig. 2. ToE (RTU) in its target environment

Internally ToE is decomposed into other parts that are relevant from the security perspective. These parts are presented in Fig. 3.

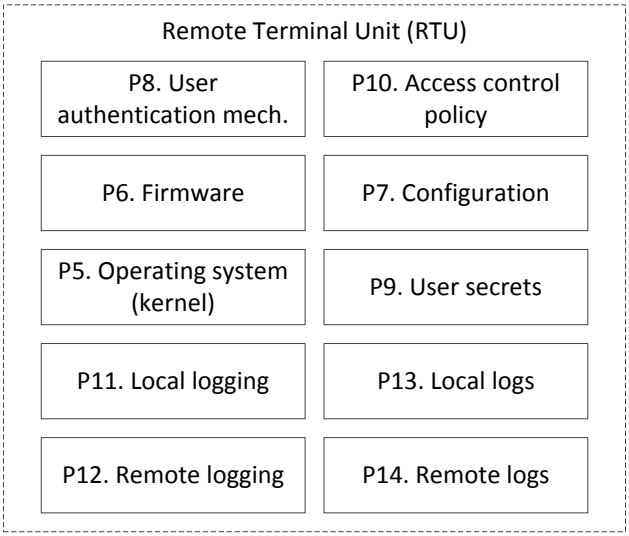

Fig. 3. ToE (RTU) internal structure

4.2 Protection Profile of RTU

We assume (based on [1]) the following main elements of a protection profile structure: 1) Description of the family of products (ToE); 2) Parts; 3) Operating conditions; 4) Critical assets; 5) Threats; 6) Protection assumptions; 7) Residual threats; 8) Security functions; 9) Threats vs security functions; 10) Mapping of security functions to security requirements.

Table 1. RTU critical assets (identified by the ‚x' symbol)

Security characteristic / *Part*	Availability	Confidentiality	Integrity	Authenticity
P5. Operating system (kernel)			x	x
P6. Firmware		x	x	x
P7. Configuration		x	x	x
P8. User authentication mechanism			x	x
P13. Local logs			x	x
P14. Remote logs			x	x

RTU parts (presented in Fig. 2 and Fig. 3) were subjected to security analysis to assess the risk related to violation of their security properties like availability,

confidentiality, integrity or authenticity. The result is a set of *critical assets* that are to be protected against attacks (a critical asset is a security property of a part that needs to be protected by security measures). Selected critical assets of ToE are given in Table 1.

The following are example threats that have been identified as being relevant, during the security analysis of the ToE.

T1. *Operating system / firmware alteration*: The attacker manages to inject and run a corrupted OS / firmware on ToE (for instance, inserts modifications without having the privilege to do so). The code injection may be temporary or permanent and this does include any unexpected or unauthorized code execution. An authorized user may attempt to install a malicious update of ToE by legitimate means.

T2. *Configuration alteration*: The attacker manages to modify, temporarily or permanently, ToE configuration.

T3. *Local logs alteration*: The attacker manages to delete or modify a local log entry without being authorized by the access control policy of ToE.

T4. *Remote logs alteration*: The attacker manages to delete or modify a remote log entry without the receiver (the component hosting the log) being able to notice it.

Table 2 presents which critical assets of ToE can be affected by the identified threats (*Av* stands for availability, *I* for integrity, *C* for confidentiality, and *Au* for authenticity). For instance, integrity of local logs (column P13) can be violated by the *Local logs alteration* threat (row T3).

Table 2. RTU critical assets affected by the threats.

Parts / Threats	P5. Operating system (kernel)	P6. Firmware	P7. Configuration	P13. Local logs	P14. Remote logs
T1. Operating system / firmware alteration	I, Au	I, Au			
T2. Configuration alteration			I, Au		
T3. Local logs alteration				I, Au	
T4. Remote logs alteration					I, Au

Critical assets are protected by Foundational Requirements (FR) that are selected to address the related threats. Table 3 presents the selection of FRs to protect the critical assets of RTU.

Table 3. Foundational Requirements assigned to critical assets of RTU

Parts / Threats	P5. Operating system (kernel)	P6. Firmware	P7. Configuration	P13. Local logs	P14. Remote logs
T1. Operating system / firmware alteration	Au: **FR1, FR2** I: **FR3**	Au: **FR1, FR2** I: **FR3**			
T2. Configuration alteration			Au: **FR1, FR2** I: **FR3**		
T3. Local logs alteration				Au: **FR1, FR2** I: **FR3, FR6**	
T4. Remote logs alteration					Au: **FR1, FR2** I: **FR3, FR6**

By grouping the critical assets assigned to the same Foundational Requirements in Table 3, we obtain the list of Security Functions (SF) of RTU. A selected SF of RTU is presented in Table 4.

Table 4. Example Security Function of RTU

Security function	Protected critical assets	Foundational requirements	Addressed Threats
SF4. User authorization in TOE functions	***Authenticity of:*** P5. Operating system (kernel) P6. Firmware P7. Configuration P8. User authentication mechanism P13. Local logs P14. Remote logs	FR 1 Identification and authentication control FR 2 Use control	T1. Operating system / firmware alteration T2. Configuration alteration T3. Authentication violation T4. Local logs alteration T5. Remote logs alteration

Then the FRs assigned to a particular Security Function can be decomposed down to the Component Requirements (CR). For instance, consider the assignment of CRs to SF4: *User authentication and authorization of ToE functions*. As shown in Table 4, SF4 has been mapped on two Foundational Requirements: FR1: *Identification and authentication control* and FR4: *Use control.* The present version of

62443-4-2 (still not formally endorsed) maps FR1 on fourteen different CRs and FR4 is mapped on thirteen different CRs. This leads to the assignment of CRs to SF4 which is presented in Table 5. From Table 4 we see that SF4 addresses only some of the critical assets of ToE and therefore not all CRs will be relevant for these critical assets. In Table 5 this is represented by listing the irrelevant CRs in gray and the relevant CRs in black.

Table 5. Example Security Function with corresponding Component Requirements

Security function	IEC 62443-4-2 requirements
SF4. User authorization in TOE functions	CR 1.1 – Human user identification and authentication CR 1.2 – Software process and device identification and authentication CR 1.3 – Account management CR 1.4 – Identifier management CR 1.5 – Authenticator management CR 1.6 – Wireless access management CR 1.7 – Strength of password-based authentication CR 1.8 – Public key infrastructure certificates CR 1.9 – Strength of public key authentication CR 1.10 – Authenticator feedback CR 1.11 – Unsuccessful login attempts CR 1.12 – System use notification CR 1.13 – Access via untrusted networks CR 1.14 – Strength of symmetric key authentication CR 2.1 – Authorization enforcement CR 2.2 – Wireless use control CR 2.3 – Use control for portable and mobile devices CR 2.4 – Mobile code CR 2.5 – Session lock CR 2.6 – Remote session termination CR 2.7 – Concurrent session control CR 2.8 – Auditable events CR 2.9 – Audit storage capacity CR 2.10 – Response to audit processing failures CR 2.11 – Timestamps CR 2.12 – Non-repudiation CR 2.13 – Use of physical diagnostic and test interfaces

5 Support for representing security requirements

Evidence-based arguments are widely used to argue about achievement of some (important) goals. For instance, an argument can justify compliance with a chosen standard or can demonstrate a critical property of a considered object, like safety of a device, security of a service and so on. An argument demonstrating the compliance is called *conformance case* whereas an argument demonstrating the selected property (such as safety, security, reliability, privacy etc.) is called *assurance case*. Recommendations on structuring assurance cases can be found in [2, 3].

Conformance case of a component belonging to a given family of products can be based on a common *argument template* derived from the related protection profile. Such template can be re-used in several concrete arguments [8]. Typically, a template contains the higher (more abstract) part of the argumentation and while converting the template to a concrete argument it is necessary to complement it with a more specific argumentation and the supporting evidence. The argument extension explicitly describes strategies of implementing the higher level security requirements in ways that are specific for a particular component. The resulting argument (containing the template, possibly some additional extended argumentation and the supporting evidence) can be subjected to the assessment, as illustrated in Fig. 4.

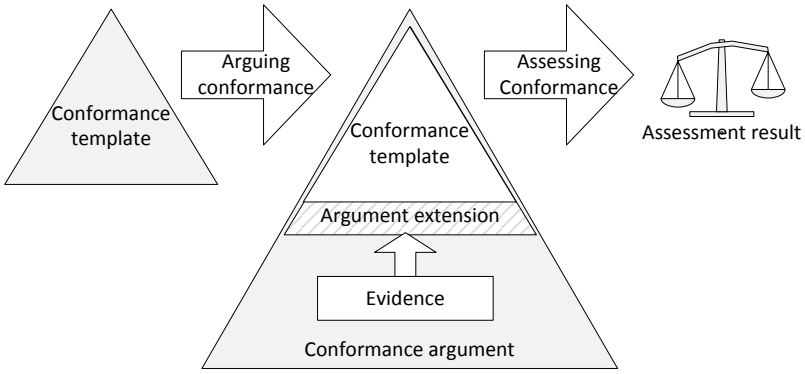

Fig. 4. Use of a conformance template while developing and assessing arguments

Templates, concrete arguments, evidence and assessments can be managed with tool support and an example of such tool is NOR-STA [4] which was used in RTU case study. NOR-STA implements TCL metamodel of evidence-based arguments compliant with ISO 15026 [3] and OMG SACM metamodel [2].

5.1 Representing conformance arguments

For a given protection profile (PP), the conformance argument demonstrates that all security functions that are relevant for this PP are effective and provide adequate protection of the related critical assets.

The following TCL elements are dedicated to representing arguments. Argument conclusion is represented by a *claim* ( ) node. A node of type *argumentation strategy* (denoted) links the claim with the corresponding premises and uses a *rationale* node (denoted) to explain and justify the inference leading from the premises to the claim. A premise is a sort of assertion and can be in particular another claim to be further justified by its own premises, a *fact* (denoted) represented by an assertion to be demonstrated by the supporting evidence, or

an *assumption* (denoted). In addition, the *reference* node (denoted) can be used to point to external documents which are integrated with the argument (for instance, to integrate external files containing evidence supporting argumentation). An auxiliary *information* node (denoted) is used to provide more structure and to explain the contents of the argumentation.

Fig. 5 illustrates how the above elements are used to represent the topmost structure of the conformance argument for RTU Protection Profile.

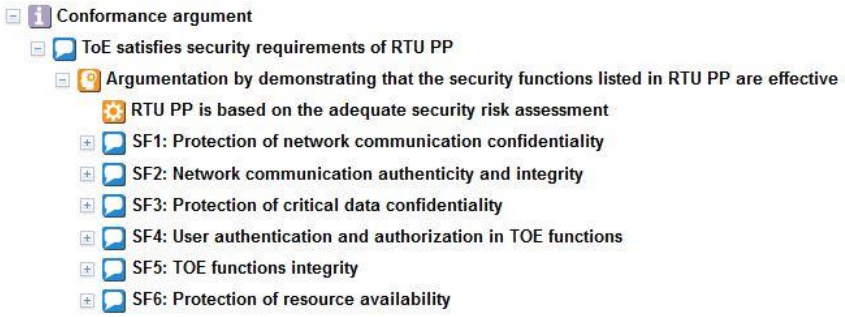

Fig. 5. A fragment of RTU Protection Profile represented in NOR-STA

5.2 From Security Functions to Component Requirements

In the RTU Protection Profile, Security Functions are assigned to critical assets and are supported by selected Foundational Requirements, as shown in Table 4. This relationship between Security Functions and the supporting Foundational Requirements is illustrated in Fig. 6 and is specified with the use of an *argumentation strategy* node and the *rationale* node which justifies the inference leading from the premises to the conclusion.

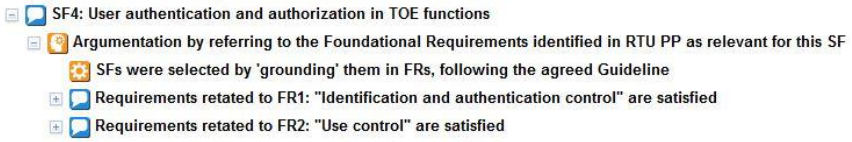

Fig. 6. An argument fragment showing how FRs support SF4 Security Function

The argumentation strategy and its rationale explain that FR1 and FR2 were identified as the support for SF4 following the Guideline of building Protection Profiles that has been agreed and accepted. Note that if such argumentation strate-

gy is considered acceptable, then to demonstrate SF4 is sufficient to demonstrate that the requirements related to FR1 and FR2 were satisfied by ToE.

Foundational Requirements (FR) selected to support a given Security Function (SF) can themselves be decomposed down to the level of Component Requirements (CR). For the RTU Protection Profile, an example of such decomposition is presented in Table 5. And this decomposition can be represented by the argument fragment shown in Fig. 7.

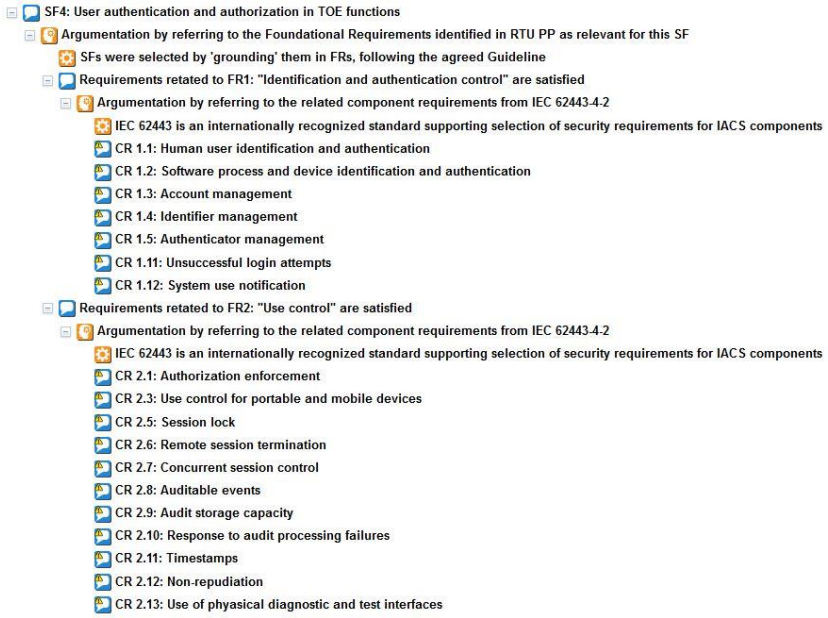

Fig. 7. SF4 supported by FRs and corresponding CRs

In Fig. 7, the same argumentation strategy is used to justify that both FR1 and FR2 will be satisfied if the Component Requirements (CRs) that support particular FR are demonstrated to be satisfied. The strategy is the same for both FRs but it has be accepted separately. If we accept the argumentation strategy for FR1, the requirement will be satisfied depending on the satisfaction of the component requirements CR1.1, CR1.2, CR1.3, CR1.4, CR1.5, CR1.11 and CR1.12.

Satisfaction of each Component Requirement (CR) can then be argued by referring to some facts that assert about component design, test results, reviews/inspections, handling procedures and so on. Fig. 8 illustrates how the satisfaction of CR 1.11 from Fig. 7 could be argued by referring to the recommended best practices of unsuccessful login handling. The facts shown in Fig. 8 are supported by the evidence which can be accessed through the corresponding reference nodes. The files containing the evidence can be stored in any external repository, for instance in the design documentation repository, test results repository and others. For instance, the evidence demonstrating that F1.11.3: *the mechanism for*

setting limit for unsuccessful logins is in place could be demonstrated by two pieces of evidence: E1.11.3.1: *an excerpt from the design documentation explaining the mechanism* and E1.11.3.2: *the report from tests verifying that the mechanism works as expected* (see Fig.8).

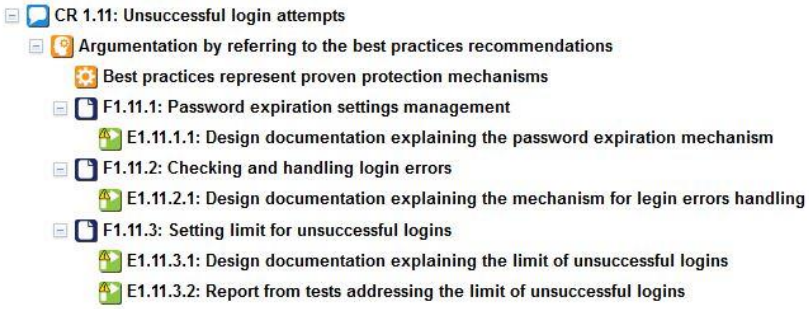

Fig. 8. Argument fragment how CR1.11 is supported by facts and evidence

6 Support for conformance assessment

Conformance arguments can be extended with the assessment data as presented in Fig. 4. In the RTU case study we used the assessment method based on Dempster-Shafer theory of evidence (the details can be found in [15]). The assessment process is explained below.

The assessor issues her/his opinion related to the acceptance/rejection of the assessed object and specifies the confidence level associated with this opinion. The assessed objects are argumentation strategies (in this case the assessor decides if he/she accepts the strategy) and facts (in this case the assessor decides to which extent a given fact has been demonstrated by the evidence supporting this fact). The assessments are expressed using linguistic values. The following values are used to express the decision of the assessor: *acceptable, tolerable, opposable, rejectable*, where *acceptable* means that the evidence fully demonstrates the assessed fact, whereas *rejectable* means that the presented evidence demonstrates the opposite. In addition, the assessor expresses confidence in her/his decision using the following linguistic values: *for_sure, with_very_high_confidence, with_high_confidence, with_low_confidence, with_very_low_confidence, lack_of_confidence*. In this case, *for sure* means that the assessor is fully confident in the decision, whereas *lack of confidence* means that he/she is fully uncertain (and in this case the decision is irrelevant). The aggregation functions of the mechanism provide for automatic propagation of these assessments into the assessments of the claims of the argumentation (for full explanation of this mechanism see [15]).

The assessment scale can be presented as an *assessment triangle* as illustrated in Fig. 9. The assessment result is a point on the assessment scale (a small shallow

token shown at the top of Fig. 9). The scale values are described on the bottom and on the right of the assessment triangle.

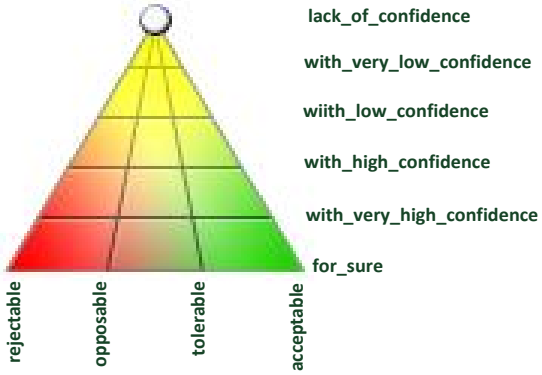

Fig. 9. The assessment scale for Dempster-Shafer method

Fig. 10 presents the result of assessing (a fragment of) the Protection Profile of RTU. The assessed element is fact F1.11.3 marked on the argument element tree. This fact is supported by evidence E1.11.3.1 and E1.11.3.2. The assessor should inspect these evidence items to decide how they support the fact and then should express his/her opinion using the assessment scale. If the evidence is not complete or ambiguous he/she may give assessment with low level of confidence. Depending on the content of the evidence the result may be acceptance or rejection.

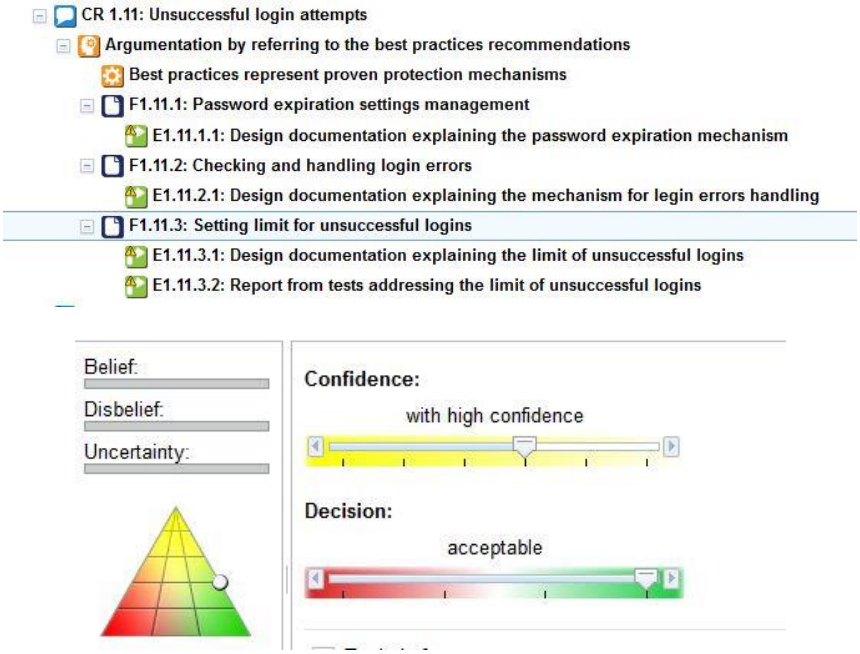

Fig. 10. Assessment of the fact F1.11.3

Assume now that the assessor has already issued his/her assessments for facts F1.11.1, F1.11.2 and F1.11.3 and in addition she/he has fully accepted the rationale of argumentation strategy for claim CR1.11. Now facts F1.11.1 and F1.11.2 are fully accepted and fact F1.11.3 has been assessed as "acceptable with high confidence" (as in Fig. 10). In such case the assessment of claim CR1.11 will be calculated automatically from the assessments of the related argumentation strategy and its premises. And the resulting assessment of CR1.11 is "acceptable with very high confidence" as shown in Fig. 11. The assessment results can also be presented with color scale: green, red and yellow colors to represent respectively acceptance, rejection and uncertainty (the colors are not visible in the black and white text of this chapter).

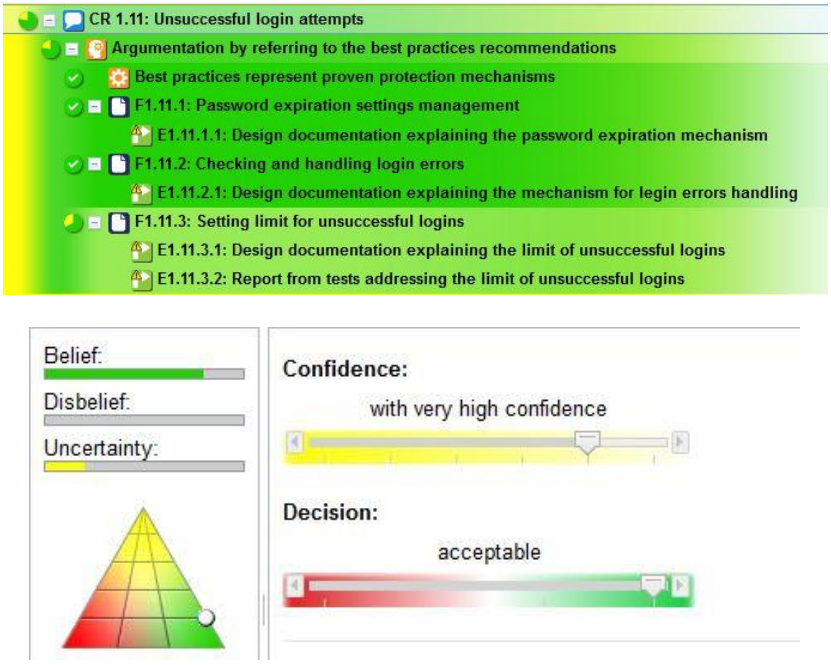

Fig. 11. Assessment of CR1.11 based on the assessments of F1.11.1, F1.11.2 and F1.11.3

To assess the whole conformance argument presented in Fig. 5 it would be necessary to assess all argumentation strategies and to assess all facts supporting the claims related to Security Functions of RTU. Note however that all those assessments are 'local' in the sense that they require that the assessor focuses on just one section of the argumentation structure (when assessing an argumentation strategy, the assessment scope covers the claim, the strategy and premises supporting it). Then, the assessment of the top claim can be calculated automatically by applying assessment aggregation rules.

7 Conclusions

In this chapter we presented a case study demonstrating how security requirements of an IEC 62443 based Protection Profile of a family of IACS components can be represented in the form of an evidence-based argument and how such argument could be used to support assessment of the compliance of an IACS component.
The proposed approach has the following advantages:

– The security requirements of the Protection Profile can be represented as an argumentation scheme (called conformance template) which can be reused for different components belonging to the same family;

- The template becomes a complete conformance argument by extending it with the argumentation that is specific for a given component and then submitting the evidence and integrating it with the argument;
- Assessment of a conformance argument can be supported by using advanced methods which provide for explicit representation of assessor's decisions as well as the uncertainty associated with these decisions;
- Automatic aggregation functions facilitate assessment of large arguments which can be encountered in practice reducing the tedious effort of calculating the overall assessment result and tracking relations between the evidence, security requirements, security functions and objectives;
- The process of template development, argument instantiation for a specific component, integration of the evidence, security assessment, and reporting and visualization of the results can be supported by a dedicated tool.

The protection profiles, related conformance arguments, the evidence supporting the argumentation and the results of conformance assessment form together a complex set of interrelated data and documentation. To process such data accurately and efficiently and to provide for scalability of such processing, it is essential to have an adequate tool support. It is important that such tools allow for seamless cooperation between security experts, component engineers and assessors.

8 Acknowledgement

This work was partially supported by a Statutory Grant of Polish Ministry of Science and Higher Education. The RTU Protection Profile presented in this chapter is based on the RTU Protection Profile originally introduced by Mr. Tomasz Szala from the Mikronika company to the NET-PL group working on validation of the IACS Components Cybersecurity Certification Framework (ICCF).

9 References

1. Paul Theron, Introduction to the European IACS components Cybersecurity Certification Framework (ICCF), DOI:10.276D/717569
2. Structured Assurance Case Metamodel (SACM), version 2.0, Object Management Group (2017)
3. ISO/IEC 15026 Systems and software engineering — Systems and software assurance
4. www.argevide.com/services/en/support/nor-sta/manual (visited 10.10.2017)
5. ISO 15408 Information technology - Security techniques - Evaluation criteria for IT security -- Part 1: Introduction and general model, ISO, 2009

6. http://www.ssi.gouv.fr/entreprise/guide/profils-de-Protection-pour-les-systemes-industriels/ (visited 7.09.2017)
7. www.argevide.com
8. Cyra L., Górski J., SCF - a Framework Supporting Achieving and Assessing Conformity with Standards, Computer Standards & Interfaces, Elsevier, 33, 2011, pp. 80-95
9. A. Ray, R. Cleaveland, Security Assurance Cases for Medical Cyber-Physical Systems, IEEE Design & Test, Volume 32, Issue 5, Oct. 2015, pp. 56-65
10. A. Finnegan, F. McCaffery, A Security Argument Pattern for Medical Device Assurance Cases, 2014 IEEE International Symposium on Software Reliability Engineering Workshops, pp. 220-225, IEEE, 2014
11. Othmane 2014] L. Othmane, P. Angin, B. Bhargava, Using Assurance Cases to Develop Iteratively Security Features Using Scrum, 2014 Ninth International Conference on Availability, Reliability and Security (ARES), IEEE, 2014
12. International Society of Automation (ISA), www.isa.org (visited 10.08.2017)
13. IEC 62443-1-1, Industrial communication networks - Network and system security - Part 1-1: Terminology, concepts and models, IEC, 2009
14. IEC 62443-4-2: Technical security requirements for IACS components
15. Cyra L., Górski J.: Support for Argument Structures Review and Assessment, Reliability Engineering and System Safety, Elsevier, 96, pp.26-37 (2011).