Maximilian Eibl, Martin Gaedke. (Hrsg.): INFORMATIK 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 987

Forensic-Related Application Security Controls for RHEL in Critical Infrastructure

Edita Bajramovic[1] and Andreas Lainer[2]

Abstract: Industrial cyber security is an avid area of research. Incident response and forensic investigations are complex activities. Due to the complexity of critical infrastructures, such as Nuclear Power Plants (NPPs), preparation is vital. Manual approaches still tend to be favored mainly because of (physical) safety assurances. The tasks and actions required and the outcomes to expect need to be documented. Application Security Controls (ASCs) are a good way to document forensic controls for which an extended model is proposed. However, ASCs must be tested further on forensic applicability and there are also considerable alternatives. In terms of possible additional security measures and to apply the gained insights, one exemplary operational Instrumentation & Control (I&C) server system is analyzed in order to derive recommendations.

Keywords: application security controls, forensics, critical infrastructure, RHEL, standards.

1 Introduction

Critical infrastructure depends on information technology during the operation. The possibility to manipulate any system of any critical infrastructure could be a major threat to many lives or physical conditions [IEC613]. This is especially true for infrastructures in the nuclear sector, for example Nuclear Power Plants (NPPs). Most existing nuclear reactors, designed 40 or 50 years ago, are managed mainly by analog Industrial and Control (I&C) systems that are not highly susceptible to modern cyberattacks [HA14]. But, newly designed reactors are managed by digital I&C systems which are highly vulnerable to cyberattacks. Examples of advanced cyber-attacks with malware are Stuxnet and Flame, both very sophisticated in design, which are targeted at specific systems to sabotage or steal information, respectively [VGA13]. Advanced targeted attacks need many organizational, financial and technical resources. Actors who are willing to make such a high effort are also capable of ensuring that traditional cyber security protection mechanism are limited in successful detection of such attacks [VGA13]. For the same reason, they are likely to be very persistent, which is why they are referred to as Advanced Persistent Threats (APTs). A lot of effort and research is being put into securing critical industrial infrastructures. However, this is very challenging because of many aspects, e.g., proprietary hardware [Fo15], complex and legacy architectures, proprietary protocols or even extinct technologies and manufacturers [FC08]. International organizations are

[1] Friedrich-Alexander-University Erlangen-Nuremberg, Department of Computer Science, Martensstrasse 3, Erlangen, 91058, edita.bajramovic@fau.de
[2] Friedrich-Alexander-University Erlangen-Nuremberg, Department of Computer Science, Martensstrasse 3, Erlangen, 91058, andreas.lainer@fau.de

doi:10.18420/in2017_101

actively working on new guidelines and standards. For example, the International Atomic Energy Agency (IAEA) started the Coordinated Research Project that focuses on incident management. Furthermore, the International Electrotechnical Commission (IEC) is now developing a new standard that focuses on security controls for NPPs in the SC 45A Subcommittee. Thorough analysis of every security incident and learning from the gathered evidence [Fo15] is very important for further improvement of existing and new nuclear I&C systems. The ability to gather and preserve evidence can be useful even before an incident occurs [Ro04]. Without adequate preparation, required evidence could be ignored, discarded or damaged at any point before, during or after an incident [Ta01]. Also, evidence may ultimately not be generated by default because the functionality to perform logging or auditing tends to be initially disabled on deployed systems, which would make additional measures such as network monitoring mandatory to help understanding the communication in case of an incident [FC08]. Generally, an effective logging system greatly assists a possible investigation with relevant logs [Ed15]. Furthermore, logging is essential as it ensures traceability. For example, one could practice Digital Forensic Readiness (DFR) to increase the usefulness of evidence and decrease the cost of doing forensics [Ta01]. Not much research is done regarding DFR in the context of critical industrial infrastructures. Forensic-related measures could, for example, be modeled as security controls [Ba17]. The fairly new international multi-part standard ISO/IEC 27034 "Application Security" postulates a concept called Application Security Control (ASC) which is a semi-formal approach aimed at specifying complex security controls [ISIE11]. The server system is expected to run Red Hat Enterprise Linux (RHEL). For this, additional guidance on the generics of RHEL was taken from the latest versions of Red Hat Enterprise Linux 7 System Administrator's Guide [RH16b] and Red Hat Enterprise Linux 7 Security Guide [RH16a]. The operating system RHEL is described in terms of forensic evidence and other technical opportunities and the model for security controls is presented and extended with forensic-related aspects.

2 Concept for Application Security

The international multi-part standard ISO/IEC 27034 "Application Security" (part of the ISO/IEC 27000 ISMS family) can assist organizations with the task of integrating security into their applications by providing concepts, principles, frameworks, components and processes [ISIE11]. The standard defines application as "any IT solution". Within the scope and context of nuclear cybersecurity, this could be, for example, a specific nuclear I&C system or multiple I&C systems [GB16]. There is a growing need for increased security because applications need to be protected against intended and unintended vulnerabilities, e.g., inherent software errors or retroactive changes to the context of an application [ISIE11]. Being able to maintain a secure development environment is an important aspect of defense strategies [So12]. A systematic approach for security integration ideally provides evidence that the information being processed by the application is protected in an appropriate way [ISIE11]. Some sub-parts have already been published as international standard and some are still at a preliminary draft stage. Part 3

of ISO/IEC 27034 describes the two overall management processes that contain all the frameworks and components. The Organization Normative Framework (ONF) Committee makes use of the ONF Management Process and is responsible for implementing and maintaining an Application Security Management Process (ASMP) [ISIE16a] while making sure it is practical for all application projects [GB16]. Each particular project is then itself responsible for implementing and utilizing a concrete ASMP to manage their security aspects, eventually resulting in an Application Normative Framework (ANF) [ISIE16a]. Fig. 1 provides an overview of the security management process and its five steps.

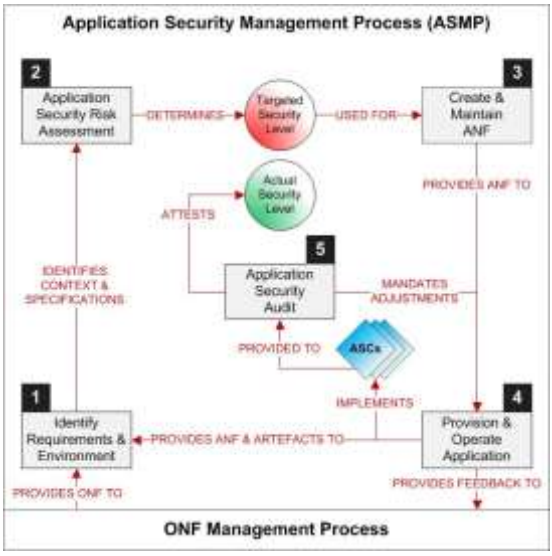

Fig. 1: Application security management process [ISIE11]

The security of an application is finally verified by the last step named application security audit. In the context of the standard, an audit takes the evidence provided by verification measures of performed security activities which have been provided as Application Security Controls (ASCs). ASCs are a semi-formal (extendable) data structure to describe complex security controls [ISIE16b]. Audits should ideally take place internally and also externally [GB16]. Parts 5 and 5-1 are of most importance to this paper because they contain requirements and recommendations on ASCs. Part 5 contains information requirements for individual ASCs. This part is already at the stage Draft International Standard (DIS), meaning that it is being inquired and targeted for publication on 2017-05-30 [ISIE17]. Thus, the contained requirements should not be subject to change. Part 5-1 contains recommendations on the formal structure for ASCs by providing graphical representations and XML schemes and can be seen as a concrete implementation [ISIE16c]. Because this part implements the requirements from Part 5, its content can also be seen as quasi-stable, although its status is still that of a Proposed Draft Technical

Specification (PDTS).

3 Application Security Controls (ASCs)

The principle of demonstrating security is supported by ASCs. Conventional controls may not be precise enough [ISIE12], requiring to create unambiguous, detailed and rich model-based security-controls. ASCs are defined as a "data structure containing a precise enumeration and description of a security activity and its associated verification measure to be performed at a specific point in the life cycle" [ISIE11]. Life-cycle in this context is defined as "evolution of a system, product, service, project or other human-made entity from conception through retirement" [ISIE11]. Applicability of ASCs includes all "processes, components, software, results, data, technologies and actors" involved in an application [ISIE11]. Possible sources for ASCs can be, for example, "standards, best practices and roles, responsibilities, and professional qualifications, technological, business, and regulatory contexts and application specifications" [ISIE11]. Processes related to ASCs can be grouped in three phases [Wa17]. The development (also creation) of ASCs is usually done by a team with a specialization in security (inside or outside staff). The distribution (also deployment) of ASCs is managed by the Organization Normative Framework (ONF) and the Application Normative Framework (ANF). The continuous maintenance (also improvement) of ASCs is finally backed by the ONF Process and the ANF Process. An organization-wide ASC library, which is part of the ONF, collects all ASCs and makes sure that a global distribution standard is established [Ba17]. ASCs are transferred from the organization library to the ANF which contains only the information required for a specific application [ISIE11]. A graphical example for an organization ASC library is shown in Fig. 2.

Application specifications, business, regulatory and technological context will yield constraints which determine the contents of the library, for example, laws to follow or compliance with standards. Key-components of an ASC are shown in Fig. 3. First, an ASC references an Application Security Life Cycle Reference Model (ASLCRM) to introduce activities and measures into existing processes. Organizations tend to already have in place life-cycle models to manage their complex applications. Such models are usually highly customized and have been in use for a long time. This is why the standards do not raise or mandate any changes to such models. The reference model is part of the ONF and related processes are, for example, "application management, application provisioning and operation, infrastructure management and application auditing" [ISIE11]. Second, there are four items that will now be briefly explained in terms of content, named Security Requirements, Targeted Security Level, Security Activity, and Verification Measure. An ASC has to state Security Requirements and targeted Level of Security (or trust) including a description on Why they are targeted or associated [ISIE16b].

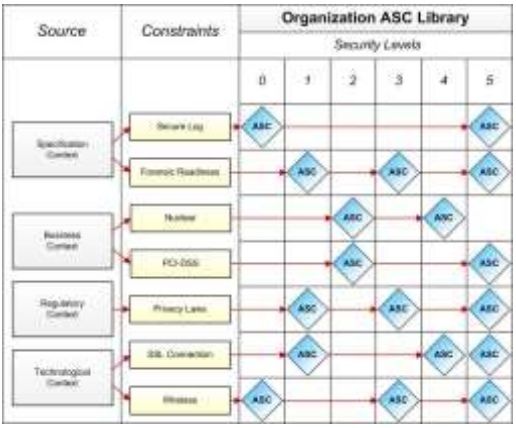

Fig. 2: ASC library [ISIE11]

Fig. 3: Key components of an ASC [ISIE11]

The descriptions of Security Activity and Verification Measure have to include additional information regarding certain questions, for example, what a high-level description as well as a detailed workflow specification for performing the activity or the measure. Additionally, ASCs can contain references to so called super-ordinate and sub-ordinate ASCs [ISIE16b]. Relations can be established via child-parent linking, making them ideal for graph visualization and analysis. One set of selected ASCs can form a package (to be exported), for example, one with all forensics-related controls. A simple graphical example for this can be seen in Fig. 4.

4 Model Specification

ASCs should be made available as XML documents [ISIE16c]. An ASC contains information on items that are usually managed on their own, for example, assets or requirements. However, the standard also recommends that ASCs should be self-contained in order to be easily exchangeable [ISIE16b]. A potential management application should

take this into account and provide a reference or link to the relevant information in order to ease management. The implementation example from the standard will now be explained in detail. Required extensions and relevant properties related to forensics are subsequently proposed and explained in the following Sections.

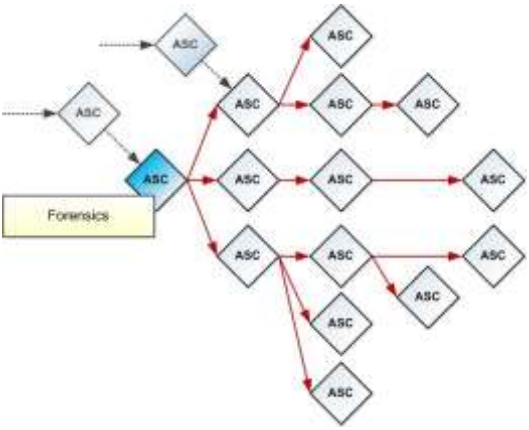

Fig. 4: Directed graph of an ASC package [ISIE11]

4.1 Structure of ASCs

The associated recommendations and information requirements from the ASC standard are implemented as a XML Schema Definition (XSD), which can be acquired from ISO/IEC directly (see [ISIE16c] or [ISIE17] for a link/contact). Optional elements have dashed frames, compound elements (no basic type) are in underline text, and important elements are in bold text. XSD files define the structure of XML documents and the contained elements and can also be used to validate instances. The header of the current and last ASC XSD version 1:0:0 from ISO/IEC can be seen below. The XSD also defines an own name space to provide references to the ASLCRM (Application Security Life-Cycle Reference Model). Each element can specify the attributes minOccurs and maxOccurs. The default value for omitted attributes is 1 for both (exactly one element). Optionality can be specified by minOccurs set to 0. By setting maxOccurs to either n 2 Z>0 or unbounded, it can be specified how many times an element can occur. ASC Package: This is the top-level element of the standard for bundling one or many related ASCs in form of a package [ISIE16c]. ASC Identification: This is the first ASC section that defines global information related to the identity of the ASC [ISIE16c]. ASC Objective: This is the second ASC section which defines key attributes of an ASC [ISIE16c]. ASC Security Activity and Verification Measure: These are the third and fourth ASC sections that contain the information that is pertinent to activity and measure [ISIE16c]. ASC Actor and Information: These complex types are references on multiple occasions from the already introduced items.

5 Forensic-Related Extensions

Activity elements (security activity and verification measure) are able to hold an unbounded amount of tasks. Forensic-related activities could be provided as an extra task in order to also make use of the ASCs as forensic-related security control. In order to distinguish the forensic tasks, the life-cycle reference that every execution moment of a task provides could be used and extended. The <ASLCRM_activity-name> element in the XSD contains and defines the stages and activities that reference the organization ASLCRM. Values such as FORENSIC_READINESS, FORENSIC_INVESTIGATION or INCIDENT_RESPONSE would need to be added there accordingly. An example implementation of a single ASC could be: The security activity configures a daemon to enable a function F. A forensic-relevant task would then contain a reference to a baseline configuration that configures the intended function F, e.g., a configuration management system. The verification measure verifies that F is enabled and functions in the intended manner. A forensic-relevant task would then contain a reference to, e.g., a logging system that contains logs related to the utilization of F. Finally, both activity elements provide lists of real example outcomes for the single commands or actions that need to be executed in order to verify their proper execution. **Life-Cycle Stage** in the identification section needs to be extended in order to incorporate the missing stages ENGINEERING (might be specified as DESIGN) and OPERATIONS. Incidents might occur during all three phases and this needs to be considered. **Parents and Children** in the identification section are to specify order and sequence of ASCs. The reference to other ASCs is provided as UID. For example, there could be a parent (detective) ASC that would contain the minimum set of properties needed for a child (detective) ASC, for example, the destination of the collected information, performed filtering, or acceptable collection delay [Wa17]. A child ASC related to logging would be derived and contain additional properties specific for logging. This would include the location of the log (could also be a buffer), permissions to access the log, format and structure of the log and the records, and notes on time stamping [Wa17]. **Requirement Type and Requirement Context** in the objective section needs to be extended to also allow for possible values like RECOMMENDATION or SECURITY_CONTROL. Although the requirement should itself also contain information on this matter, this would support the principle of self-contained ASCs. **Assigned Levels of Trust** in the objective section refers to an either outside or inside definition of the level of trust. This resembles the graded approach with security degrees or security levels. Examples are the degrees from ISO/IEC 62645 (S1, S2, S3 and BR) [IEC614], or an own grading and definition (e.g., 1 to 10). The levels need to be mapped in order to be interchangeable [Wa17] (see <levels-of-trust-range> element). **Conditions Type** of the conditions in the objective section may need to be extended. For example, a control that is “destructive” in terms of volatile evidence would need to contain a precondition to only execute the particular control after volatile evidence has already been collected, e.g., AFTER_VOLATILE_EVIDENCE. A list of threat assumptions that are aimed to be mitigated should also be included here in the conditions. **Information Items** of the target information in the activity synopsis contains precise information on information items. Both security activity and verification measure of an ASC are likely to

have the same scope and to affect the same processes and assets. Thus, the current implementation of the format results in an exact copy of this information within both activities. The reasons for this are, so far, unknown and this should be clarified first before considering to implement concrete ASC. However, the standard is still in the making which might explain this potential flaw. Also, the addition of an <uid> element should be considered to ease linking. **Artifact Type** of the outcome of a task in the activity specification needs to be extended with data types (e.g., DT_4_CONFIGURATION) in order to be forensically applicable, or at least with a type EVIDENCE.

5.1 Linking to Assets and Requirements

ASCs have to describe the scope and the affected information items, e.g., processes and assets. The existence of a proper asset management is an accepted precondition for conducting forensic investigations [Li16]. For example, ISO/IEC 27002 [ISIE13b] contains security controls regarding asset management (A.8) and their inventory, ownership, acceptable use and return (A.8.1) as well as the classification of information (A.8.2). The asset modeling or documentation process could be supported with 2D or 3D models of facilities [Ba17], making it more accessible and visual to support, for example, security zone definition, security control assignment, and security risk assessment [Se16]. While the inventory of assets tends to be done manually in some cases, an additional automatic asset identification should be used in order to improve the efficiency, e.g., on lower security zones [Li16]. What should at least be included in the configuration of an ASC is the exact position of the asset, e.g., within the building and within the rack [Wa17]. The identification of assets is typically utilized via some unique id or UID (at least unique within the organization) that should be part of the ASC. Additional information may then be accessed via this link. Assets are usually managed with own processes and applications. In case of a 3D or 2D model of the plant, the links should be bidirectional from both management applications [Se16]. Also, ASCs provide information on the requirements the ASC originates from. A less strict form of requirements are recommendations or security controls (also see proposed extension in the previous Section 3.2.2). Requirements are typically also managed in an own application and identified with an UID (used to link to additional information). For example, ISO/IEC 27001 [ISIE13a] contains the requirement “The organization shall determine its requirements for information security and the continuity of information security management in adverse situation, e.g., during a crisis or disaster” for the planning of information security. The recommendations regarding the management processes and applications for assets are also true for requirements.

5.2 Support on Attack Tree Generation

After a time-line of events has been successfully reconstructed, Tu et al. [Tu12] suggest to compose the meta-data of the associated attack operations into graphical nodes that are later developed into a larger augmented attack tree. As a result of this development, an

evidence tree for each considered attack is created [Ba17]. Fig. 6 shows a simple example attack tree for modifying a water pump I&C system in the form of a Bayesian Network, or Bayesian Attack Graph in this case. According to the assigned security level and the security zone where the water pump is located, certain ASCs related to forensics are selected and packed together [Ba17]. The attack tree shows various ASCs packages for different tasks. In this case, the first ASC package specifies which overall requirements the logs must have. It then descents and becomes more concrete with an ASC package with all internal logging measures, an ASC package with all external logging measures, a forensic-related ASC package on how to collect the logs (evidence), and finally an ASC package on where to store the logs [Ba17]. Another depicted package in Fig. 6 is concerned with configuration backup measures. The package approach supports that all activities are performed and none are bypassed in order to not endanger evidence [Ba17]. Also, this approach makes sure that only the relevant information is provided, for example, for a certain security level [Ba17]. ASCs need to describe tasks and the included actions to be performed, and the outcomes to expect [ISIE16b].

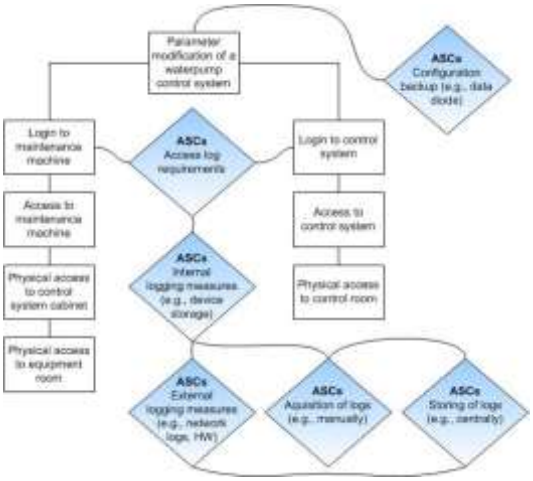

Fig. 5: Example attack tree and applied ASCs

The risk assessment process could be greatly assisted by providing information that may allow to generate attack paths or attack trees [Se16]. Information that may be used to show in the single nodes is, for example, meta-data that includes (but is not limited to) the log name, the format, the location, any timestamps, and all associated security features [Tu12]. The information that is especially contained in the following elements of an ASC may come in very useful: <parents> and <children> (edges), <target-information> (assets), <activity-complexity> (probabilities), <action-list> (possibilities) and <outcome> (evidence).

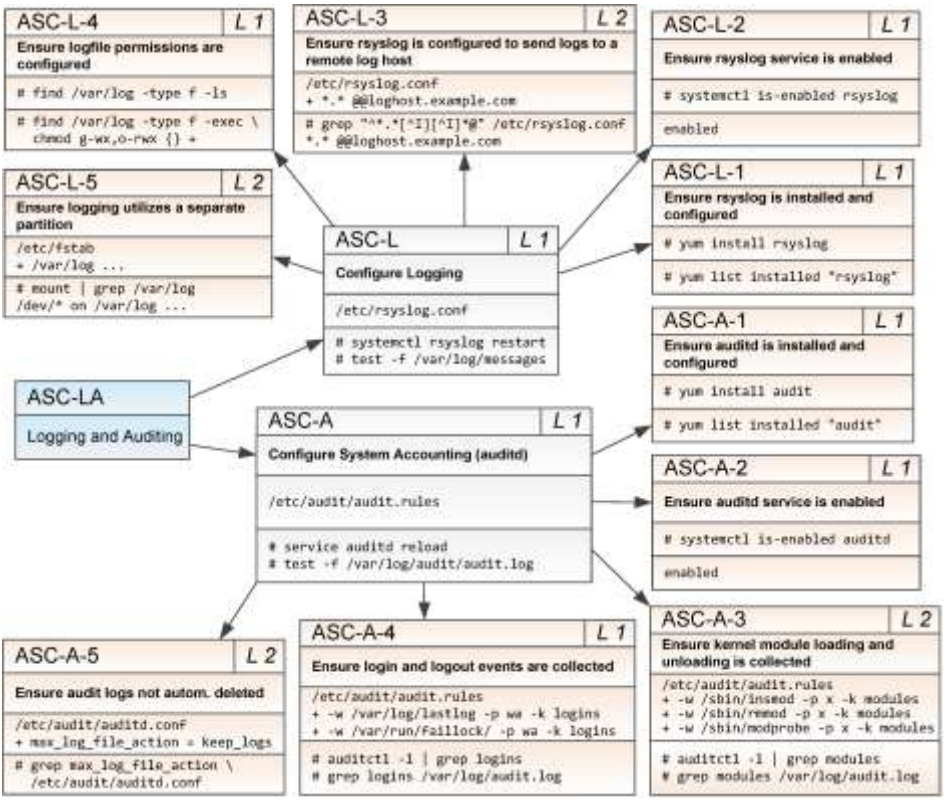

Fig. 6: Example logging and auditing ASC package for RHEL

6 Management Application Prototype for ASCs

A selection of the above list from the CIS guide has been taken and further detailed in Fig. 7 with information regarding the security level, the security activity, and the verification measure. This aids as a first concrete example on how forensic-related ASCs could be implemented. In order to demonstrate the technological values of Application Security Controls (ASCs), a small management application prototype for ASCs was also developed, created with TypeScript and Node.js. A picture of the web-application is provided in Fig. 8. However, the application is still very limited and was so far only created to show the visual aspects of ASCs.

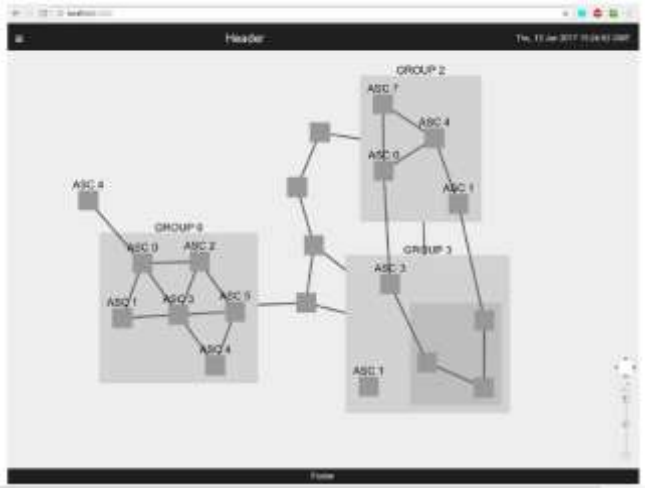

Fig. 7: Management application prototype for ASCs

7 Conclusion

The pervasive computerization and the incorporation of commercial IT products in NPPs gradually exposed the technology to risks initially not designed to handle [Fo15]. There are many good security measures from domains like IT that can be adopted and further developed to cope with the risks. Proven solutions should be favored in order to not accidentally and additionally increase the attack surface due to poor and ill-conceived solutions. Also, there is the very important requirement that security measures for a system must not have impact on its reliability [IEC616]. Non-intrusive solutions such as network security monitoring systems are well suited to fulfil this requirement because they do not disrupt time-critical communications and processes [Ng16]. Furthermore, security measures need to be incorporated into all life-cycles of a product [ISIE11]. Regarding DFIR, adequate preparation and response planning is key. Due to the fact that both forensic investigation and incident response are highly knowledge based, good ways to document and visualize this knowledge are needed, one example being Application Security Controls (ASCs). However, the forensic applicability of ASCs needs to be further tested due to the discussed limitations. Documentation is important to avoid a potentials loss of information due to, for example, resigning employees at vendors or operators. The common mistake of not taking extensive notes during an incident investigation [Kn15] should also be avoided. All these approaches combined might finally allow for a successful, optimized and cost-efficient analysis of an ICS compromise [Fo15].

References

[Ba17] Bajramovic, E. et.al.: Planning the Selection and Assignment of Security Forensics Countermeasures. In: Proc. of ICONE 25, China, 2017.

[Ed15] Eden P. et.al.: A Forensic Taxonomy of SCADA Systems and Approach to Incident Response. In: Proc. of ICS-CSR 2015, Germany, 2015.

[FC08] Fabro M.; Cornelius, E.: Recommended Practice: Creating Cyber Forensics Plans for Control Systems. In: Control Systems Security Program. US DHS National Cyber Security Division (NCSD), 2008.

[Fo15] Folkerth, L.; Forensic Analysis of Industrial Control Systems. In: InfoSec Reading Room. SANS Institute, 2015.

[GB16] Gupta, D; Bajramovic, E.: Security Culture for Nuclear Facilities. In: Proc. of NuSTEC 2016, Malaysia, 2016.

[HA14] Holt, M.; Andrews, A.: Nuclear Power Plant Security and Vulnerabilities. Congressional Research Service, 2014.

[IEC613] IEC 62443-3-3, Industrial comm. networks – Network and system security – Part 3-3: System sec. reqs. and sec. levels. IEC, 2013.

[IEC614] IEC 62645, NPPs – I&C–Req. for sec. programmes for computer-based systems, 2014.

[IEC616] IEC 62859, NPPs – I&C Systems – Reqs. for coordinating safety and cybersecurity. IEC, 2016.

[ISIE11] ISO/IEC 27034-1, IT – Sec. tech. – App. Sec. – Part 1: Overview and concepts. ISO/IEC, 2011.

[ISIE12] ISO/IEC 27000, IT – Sec. tech. – ISMS – Overview, ISO/IEC, 2012.

[ISIE17] ISO/IEC 27034-5, www.iso.org/iso/catalogue_detail.htm?csnumber=55585, accessed: 2017/01/24.

[ISIE13a] ISO/IEC 27001, IT – Sec. tech. – ISMS – Requirements. ISO/IEC, 2013.

[ISIE13b] ISO/IEC 27002, IT – Sec. tech. – Code of practice for information security controls. ISO/IEC, 2013.

[ISIE16a] ISO/IEC DIS 27034-3, IT – Sec. tech. – App. Sec. – Part 3: Application security management process. ISO/IEC, 2016.

[ISIE16b] ISO/IEC CD 27034-5, IT – Sec. tech. – App. Sec. – Part 5: Protocols and application security controls data structure. ISO/IEC, 2016.

[ISIE16c] ISO/IEC PDTS 27034-5-1, IT – Sec. tech. – App. Sec. – Part 5-1: Protocols and app. security controls data structure – XML Schemas. ISO/IEC, 2016.

[Kn15] Knowles, B. S.: DFIR Analysis and Reporting Improvements with Scientific Notebook Software. In: Penetration Testing. SANS Institute, 2015.

[Li16] Li, J. et.al.: Graded Security Forensics Readiness of SCADA Systems. In: Proc. of INFORMATIK 2016, Austria, 2016.

[Ng16] Nguyen, T. D.: Network forensics lessons for industrial control systems. Tech. Rep. Naval Postgraduate School (NPS), 2016.

[RH16a] Red Hat Enterprise Linux 7 Security Guide. Revision 1-23, Red Hat Inc, 2016.

[RH16b] Red Hat Enterprise Linux 7 System Administrator's Guide. Revision 0.14-9. Red Hat Inc, 2016.

[Ro04] Rowlingson. R.: A Ten Step Process for Forensic Readiness. In: IJDE, 2004.

[Se16] Seibt S. et.al.: 3D Modeling of Selected Assets Security Zones and Conduits. In: Proc. of INFORMATIK 2016. Austria, 2016.

[So12] Song J.-G. et.al. A cyber security risk assessment for the design of I&C systems in nuclear power plants. In: Nuclear Eng. and Tech. 44.8, 2012.

[Ta01] Tan. J.: Forensic Readiness. Inc. Cambridge, Massachusetts, 2001.

[Tu12] Tu, M. et.al.: Forensic Evidence Identification and Modeling for Attacks against a Simulated Online Business Information System. In: Journal of Digital Forensics, Security and Law 7.4, 2012.

[VGA13] Virvilis N.; Gritzalis, D.; Apostolopoulos, T.: Trusted Computing vs. Advanced Persistent Threats: Can a Defender Win This Game? In: Proc. of the 10th ATC 2013. Italy, 2013.

[Wa17] Waedt K. et.al.: Development, Distribution and Maintenance of Application Security Controls for Nuclear. In: Proc. of ICONE 25. China, 2017.