ARCHITECTURAL MODEL FOR INFORMATION SECURITY ANALYSIS OF CRITICAL INFORMATION INFRASTRUCTURES

Zhendong Ma, Paul Smith, and Florian Skopik

Safety & Security Department
Austrian Institute of Technology
{firstname.lastname}@ait.ac.at

Keywords

Critical Information Infrastructure, information security, system architecture, reference model

Abstract

Critical Information Infrastructures (CII) are computer systems and networks that support and control operations of many critical infrastructures that our society depends on, such as power plants, electrical grids, and water and waste facilities. Since the operations of CII also effect physical world, they are a good example of large-scale, critical cyber-physical systems. In recent years, CII become an attractive target for cyber attacks and the potential impact of a successful attack could lead to disastrous consequences in the physical world. Thus ensuring the security of CII is of vital importance. A fundamental prerequisite to secure a CII system is a clear understanding and a consistent view of its architecture. However, because of the complexity and scale, this is challenging to acquire. In this paper, we propose a layered architectural view for CII, which aims at building a common ground among stakeholders and supporting the implementation of information security management processes. In order to manage the complexity and scale, we define four interrelated architectural layers, and use the concept of viewpoints to focus on a subset of the system. We indicate the applicability of our approach in the context of CII security analysis.

1. Introduction

Critical Information Infrastructures (CII) are computer systems that monitor and control industrial facilities and processes. CII are used in many critical infrastructures such as power grid, traffic management, gas and water facilities. CII consist of conventional IT systems and industrial control systems such as Supervisory Control and Data Acquisition (SCADA) systems, in which computer applications as well as human operators in the control centre collect measurements from remotely connected sensors and send commands to actuators in the field according to predefined process models. Consequently, CII extend the activities in the cyber space to the physical world to form large-scale and complex cyber-physical systems (CPS). Because of the critical nature of the physical devices and services they monitor and control, CII are an attractive target for cyber attacks. Since CII become increasingly interconnected and tend to use Commercial-Off-The-Shelf (COTS) IT products as well as open standards, they have to face the same vulnerabilities and threats that plague conventional IT systems. In addition, connectivity and the use of generalpurpose hardware and software products make it easier for an attacker to understand and search for weakness in the system. Stuxnet [R. Langner (2011)] is a well-exposed wakeup call on the imminent danger facing CII. Identifying the vulnerabilities and threats and protecting CII against cyber attacks is of vital importance. However, due to their characteristics this is not straightforward. CII are complex, heterogeneous, and large-scale Besides, CII have very strict real-time requirements, and the lifecycle of components in CII tend to be much longer than normal IT systems. This makes changing and hardening the infrastructure for security purposes difficult. Despite this being an acknowledged problem, securing CII continues to be a significant challenge [V. M. Igure, S. A. Laughter, & R. D. Williams (2006), McAfee (2011), Symantec (2011)].

Clearly, to secure CII a systematic approach must be taken. A fundamental building block in the information security management process is *security analysis*, which aims at identifying assets, vulnerabilities, and the associated threats and potential attacks. A prerequisite of security analysis is to gain a systematic and comprehensive understanding of the CII under consideration. Due to their scale, complexity, and heterogeneity, a consistent view rarely exist. To this end, we propose a novel way to organize and describe CII systems and their environments and contexts using architectural layers. As often required in software design-level risk analysis, the layered architectural view is envisioned to assist security analysis by slicing and organizing CII to different technological concerns and layers of abstraction in order to build up a consistent "forest-level" view of the target system at a reasonably high level.

In the following: Section 2 describes related work and the motivation for a novel architectural view. Section 3 describes our approach to establish the architectural view. Section 4 discusses the applicability in the context of a set of security-related processes. Section 5 concludes the paper.

2. Related work

A consistent architectural view establishes a framework for understanding the target system, the entities within the system and their relationships, and system environments. When undertaking security analysis that involves different organizations and personnel, a reference architectural view helps to maintain consistency and common consensus among the participants. Common approaches so far are to model CII system in accordance with its network topology. For example, a typical architecture for power grid includes field devices connected to the SCADA network, which is connected to a corporate network [Pacific Northwest National Laboratory and U.S. Department of Energy (2006)]. Such a topological view of the architecture is also used in [K. Stouffer, J. Falco, & K. Kent (2011), US-CERT]. In our approach, details of the network infrastructure are captured in a distinct layer that describes how data may flow in a CII; additionally, we also capture other aspects that are equally important from a security perspective.

Other approaches focus on the software services within CII. The VIKING reference architecture [VIKING project (2010)] models services and data flow, and their relationship to the network topology. An architecture meta-model includes three components: data-flow, service, and zone. In our approach, these aspects are modelled across different layers, making it more straightforward to identify and analyse the security aspects of each of the entities. Coupled with viewpoints, our layers can be collapse to consider multiple layers in security analysis.

The ability to identify and classify interdependencies within CII is important for security. Berg and Stamp [M. Berg and J. Stamp (2005)] proposed a system reference architecture by applying Object-Role Modelling [T. Halpin] in order to model data, functionality and internal interdependencies in

CII. This approach is based on a mixture of function and network topology. As mentioned earlier, we separate infrastructure and networking aspects into distinct layers in order to make their analysis more approachable. Using viewpoints, we can also collapse the layers in case of carrying out the object role modelling proposed by Berg and Stamp.

A slightly different approach is to first identify SCADA devices and then logically group their functions into abstraction layers. The ISA S99 standard [ISA99 (2007)] proposes to create a reference architecture from the entities identified as assets within an organization, and build the architecture model according to the specifics of the organization. ISA S99 maps the functional components of a CII system into five architectural levels: physical process, local or basic control, supervisory control, operations management, and enterprise systems. Our approach follows the same principle, i.e., we propose to first build an asset layer as the basis for other layers. Based on existing work, it can be observed that there is no consensus on how to model CII architectures, including what should be modelled. However, system architectures tend to be modelled to reflect network segmentation. There is no single solution that provides us with a comprehensive and adaptable view of CII system architecture for applying security analysis.

3. Architecture model

A reference architecture should capture the essence of the architecture of a collection of systems. Our architectural view is structured into four layers, which can be considered in the context of arbitrary viewpoints. This arrangement is summarized in Figure 1. Each layer is intended to group system components and aspects for security analysis. A layer in the architecture consists of entities that are typically considered discrete -- for example, the communication layer includes aspects from layer two and three of the OSI reference model, and the asset layer describes physical and logical entities. Since each CII system is unique, e.g., using a range of components and subsystems from different vendors configured in different ways, it is impossible to have an architecture model that captures all peculiarities of different CII systems working in different organizations and domains. Therefore, our proposal here is meant to be an architectural template, based on which specific CII system architectural views can be derived and instantiated.

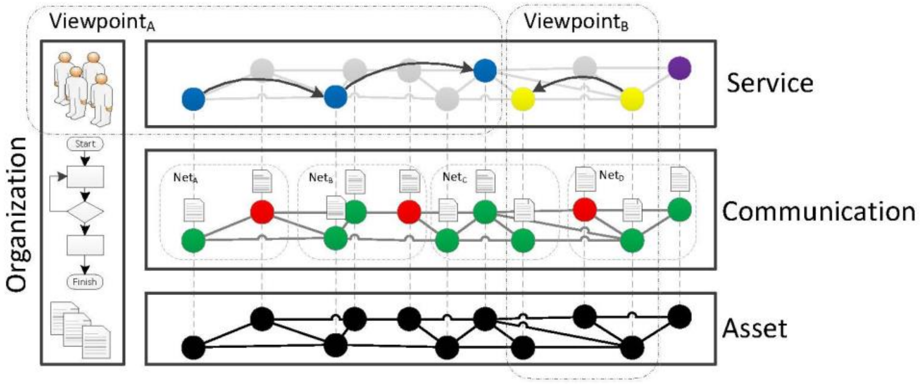

Figure 1 Layered CII architectural view

To further manage the complexity and scale of CII, we make use of the concept of viewpoints. A viewpoint is *a technique for abstraction using a selected set of architectural concepts and structuring rules, in order to focus on particular concerns within that system* [G. Muller (2011)]. A viewpoint is an arbitrary view of a CII that focuses on a subset of the system. A viewpoint can include system components from the same architectural layer as well as those from different layers.

The asset layer includes entities such as hardware, software, and data of a CII system that is usually considered as the IT asset of an organization. Hardware in CII are physical devices as well as the communication links that connect them. Devices in CII can include those typically associated with enterprise networks, such as workstations and servers, and those that are related to SCADA systems, such as field devices, including Intelligent Electronic Devices (IDEs), Remote Terminal Unit (RTUs), Programmable Logic Controller (PLCs) and Distributed Control Systems (DCS). The field devices monitor meter readings and equipment status and control end devices such as sensors and actuators. Usually, the wired and wireless communication links connect the devices into the following topologies: geographically distributed field devices are connected over various communication links (e.g., dial-up telephone, leased line, power line, radio, and Wide Area Network (WAN)) to control centres in SCADA networks; the SCADA network is connected to a company's corporate network, and the corporate network is further connected to the Internet; firewalls are used to separate and protect different networks. Software includes operating systems, databases, and application software. Data is generated and processed by hardware and software components in SCADA systems. In the asset layer, the software and data are associated with specific hardware. Figure 2 illustrates an example of asset layer.

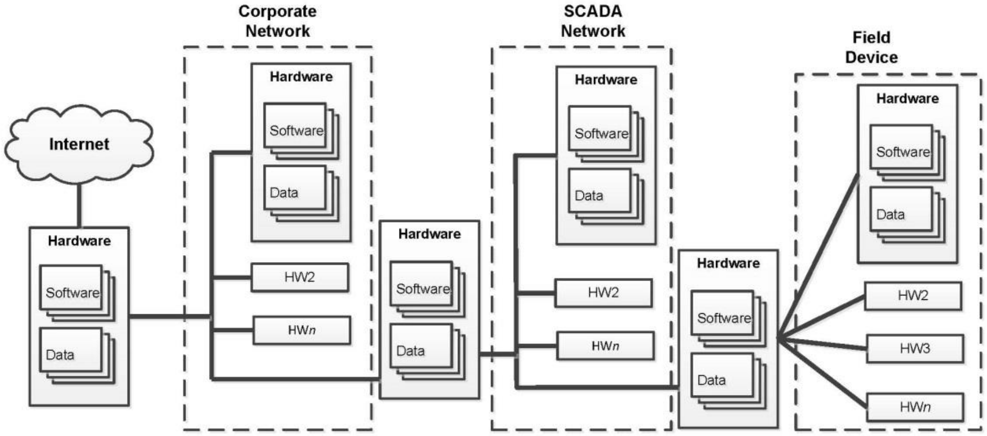

Figure 2 Asset layer

Components in the asset layer should be relatively straightforward to identify. For example, a critical infrastructure asset owner typically has detailed information on each of the hardware devices and how they are connected -- e.g., maintained in an asset management system -- as well as the software installed on that hardware and the data exchanged at the I/O ports or APIs. The components can be specified using common IT asset specification methods like the "Specification for asset identification" [J. Wunder, A. Halbardier, & D. Waltermire (2011)], which defines a data model with asset types such as software, database, network, and service etc.

The way data can be transmitted and the means of realizing these data flows, e.g., using various protocols and services, is important for CII security -- the communication layer aims to describe this. This understanding can be applied when carrying out a threat analysis to determine the reachability of critical assets from remote networks, both internal and external. Furthermore, understanding which protocols are being used can identify vulnerabilities in the CII system. Building on top of the entities described in the asset layer, the characteristics of three main classes of entity are described in this layer: (1) communication enablers, (2) communication inhibitors, and (3) communication end points, as shown in Figure 3.

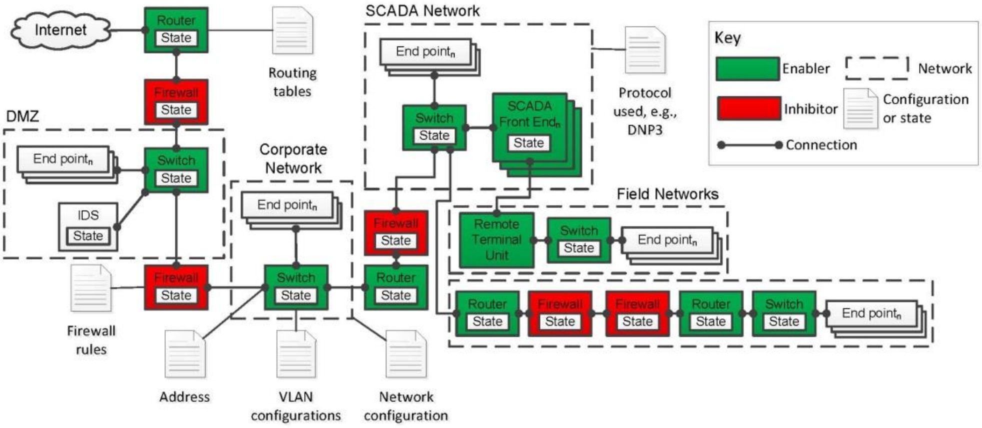

Figure 3 Communication layer

Communication enablers include devices such as network hubs, switches and routers, e.g., a further form of enabler includes the means of interconnecting these devices, physically or virtually, e.g., with the use of Virtual Local Area Networks (VLANs). In short, communication enablers describe and implement how data should flow through the CII system. In contrast to the various communication enablers, inhibitors curb the flow of data through the CII system, and typically take the form of so-called middle-boxes, such as firewalls and Network Address Translation (NAT) devices. An intrinsic communication inhibitor in CII systems can come from the heterogeneity of the protocols that are used. For example, TCP/IP is the ubiquitous protocol arrangement in enterprise networks; whereas in control networks a wide-range of protocols are used, such as DNP3 and Modbus. The communication end points are entities sending or receiving data.

The service layer models the software services, applications, and functions in CII systems and the data exchanged among the services. We use the term “service” in a broad sense to denote software components that encapsulate and provide certain functionality. Consequently, an application can be considered as a service because it provides certain functionalities to other applications and the system. Likewise, databases, authentication servers, Web servers and application servers are also considered as services. A service can be a composition of standalone services which provides customized functionalities and business applications. A service can be implemented and deployed using numerous techniques, ranging from low-level embedded systems components to application software and flexible service mashups and orchestration engines. The data flows models the data exchanges among the services.

An example service layer is illustrated in Figure 4. Services like sensor and actuator send measurement data and receive process commands to and from SCADA server through front end. HMI services are the interfaces of the operators to the SCADA server. Historian stores the historical data. A web browser in the corporate network can access these data. The SCADA application server provides various control centre applications such as power projects and overview monitoring. A Geographic Information System (GIS) service provides GIS data to data engineering server, which defines data structures and views for various services in the SCADA network. Communication server allows remote client to have terminal access to SCADA server for faster and more efficient maintenance work and information acquiring.

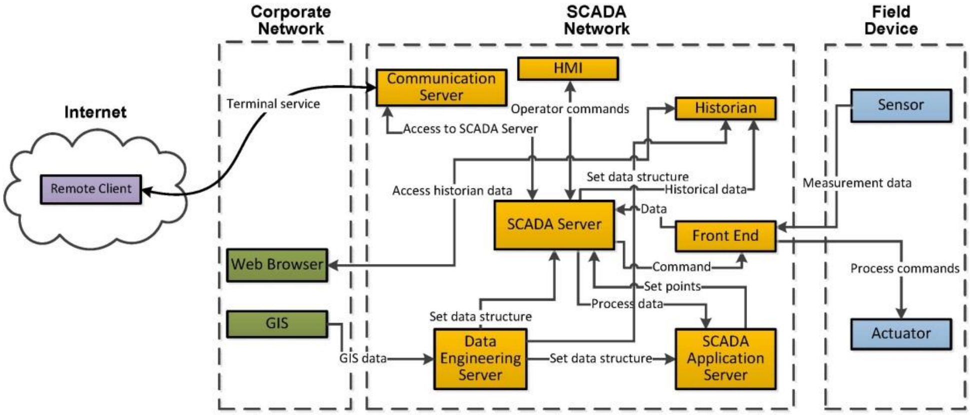

Figure 4 Service layer

Adapting some of the Service Oriented Architecture (SOA) terminology, we can describe a service with the following attributes: (1) Service Descriptions define the capabilities and functional properties of a service, as well as the communication endpoint and operations supported by the service; (2) Interactions and Data Contracts, which define the schema used for exchanged messages as well as the protocol of interaction; (3) Fault Handling Procedures provide information in case of failures and undeliverable services; (4) Service Level Agreements hold important information about guaranteed quality criteria, such as availability, accuracy, responsiveness and so on. While these definitions are more feasible for business software in cooperate networks, the same concepts are also applicable to services in SCADA networks. For instance, in order to use a deployed firmware-controlled sensor, one needs to know the protocol and applicable messages to interact with this sensor; furthermore, what operations this sensor provides (type of measurement, value ranges, operating modes), how faults are signalized, and what level of service in terms of availability or accuracy this sensor guarantees.

The organization layer consists of relevant people and their activities, as described by organizational processes and policies. As an orthogonal layer, the entities in the organization layer are related in different ways to the other layers. This includes end-users of services provisioned by CII systems as well as maintenance personnel keeping the whole infrastructure up and running. People on an organization layer have predefined privileges (e.g., roles and access rights) on peer objects, including hardware, network configurations, services and data entities. A majority of an organization's activities can be described by business processes. Business processes can potentially span numerous departments across organization boundaries. Business requirements and corporate rules affect the execution of business processes, i.e., the order and context of tasks being performed. Furthermore, in most processes coordination through and intervention of humans is required, which makes people not only system end-users, but integral parts of the whole system architecture. When executing such processes, predefined tasks are performed by different stakeholders in series. This requires the delegation of privileges among people, for instance, the ownership of data objects depending on the current task. Finally, the execution of processes is influenced by security policies and guidelines, especially, how close they are lived and applied in the business context. Thorough monitoring of user actions and review with respect to these security-relevant artifacts can reveal weaknesses in corporate procedures.

We regard security policies, which describe security administration rules and enforcement hierarchy, as an integral part of the organization layer. Security policies include those for general IT systems, such as information security and risk management policies, as well as specific policies for

CII systems such as platform security, communication security, and application security policies [D. Kilman & J. Stamp (2005)].

As mentioned earlier, viewpoints are intended to provide a focused view of a subset of the architectural layers. A viewpoint can be *horizontally* aligned to a layer, or *vertically* intersect different layers. Viewpoints can be arbitrarily defined based on the security analysis that is being carried out. For example, a security team may wish to validate how an organizational level security policy is implemented from a technology and processes perspective. To do this they may define a set of viewpoints from the perspective of the different security policies under examination, which cuts across the organizational, service and communication layers, for example. This could reveal how a security policy is implemented in service access control mechanisms and firewall policies. A viewpoint can also be defined that only intersects a subset of different layers. For example, to analysis how organisational business process impacts system security, a viewpoint can include a part in the organizational layer and another part of role-based access policies in the service layer.

4. Security analysis

The precise nature of the instantiation of the architectural view, e.g., which entities will be enumerated and their attributes in accordance with the architectural template, will depend on its application to the security analysis of specific CII systems. A security team can make use of abstract representations of the architectural view to manage the complexity and scale of the system, and to introduce some automation to the process. We foresee a number of applications of our reference architecture. For example, we can use the framework from Schaeffer [A. Schaeffer-Filho, P. Smith, A. Mauthe, D. Hutchison, Y. Yu, & M. Fry (2012)] for developing and evaluating so-called resilience patterns that describe the configuration of various mechanisms, e.g., firewalls and anomaly detection systems that can be used to detect and mitigate well-known attacks, such as DDoS attacks. Their framework makes use of simulations to evaluate candidate resilience patterns; our reference architecture could be used to support the realisation of simulation models in this context. In a more formal way, we can apply the layered architectural view for attack modelling in smart grid proposed by Chen [T. M. Chen, J. C. Sanchez-Aarnoutse, & J. Buford (2011)] that makes use of Petri-nets. In their approach, “low-level” Petri-nets are created by domain experts that describe attacks in detail for sub-domains of a smart grid, e.g., attacks on smart meters. Then the low-level attack descriptions are merged with “high-level” Petri-nets that abstract details of an attack, and focus on important places, i.e., attack states. Common places in the two types of Petri-net are merged by identifying the transitions and places described by a common model description language. Using the layered architectural view, we can support this attack modelling approach -- for example, the systematic identification of low-level Petri-nets could be done on a per-layer basis, and viewpoints could be defined that identify places and their attributes across multiple layers. Furthermore, the lexicon of the model description language could be derived from the attributes contained in an instantiation of our reference architecture.

5. Conclusion

CII systems are the IT backbone of many critical infrastructures. Security analysis that identifies vulnerabilities, threats and attacks is an important task for securing and protecting critical infrastructures against cyber attacks. Establishing a consistent architectural view of the target system should be the first step in any security analysis. In this paper, we proposed a layered architectural view to support the implementation of security analysis. As a novel way to organize and describe architectural information and to manage complexity and scale, we model CII systems in four layers: asset, communication, service, and organization layer. In addition, we introduced the concept of architectural viewpoints, which enables us to have a focused view on a subset of the system of interest during security analysis. We are aware that it is a challenging task to enforce a unanimous view on CII architecture among various stakeholders. The proposed architectural view is an attempt to establish and maintain a consistent view on the system architecture during security analysis. With abstract and focused presentations in the architectural view, we envision that more theoretical and formal methods, as well as automation techniques can be developed and applied for security analysis in CII.

Acknowledgement

This work was supported by the EU FP7 research project PRECYSE (Prevention, protection and REaction to CYber attackS to critical infrastructurEs, FP7-SEC-2012-1-285181).

References

A. Schaeffer-Filho, P. Smith, A. Mauthe, D. Hutchison, Y. Yu, & M. Fry (2012), A framework for the design and evaluation of network resilience management, NOMS, 401--408.

D. Kilman & J. Stamp (2005), Framework for SCADA Security Policy, Sandia National Laboratories report SAND2005-1002C.

G. Muller (2011), A Reference Architecture Primer, version 0.6.

ISA99 (2007), Security for Industrial Automation and Control Systems-Part 1: Terminology, Concepts and Models, ANSI/ISA-99.00.01-2007.

J. Wunder, A. Halbardier, & D. Waltermire (2011), Specification for asset identification 1.1," NIST Interagency Report 7693.

K. Stouffer, J. Falco, & K. Kent (2011), Guide to Industrial Control Systems (ICS) Security, NIST SP 800-82.

McAfee (2011), In the Dark: Crucial Industries Confront Cyberattacks.

M. Berg and J. Stamp (2005), A Reference Model for Control and Automation Systems in Electric Power, Sandia National Laboratories report 2005-1000C.

Pacific Northwest National Laboratory and U.S. Dept. of Energy (2006), The role of authenticated communications for electric power distribution, Beyond SCADA: Networked Embedded Control for Cyber Physical Systems.

Symantec (2011), Symantec Critical Infrastructure Protection Survey.

R. Langner (2011), Stuxnet: Dissecting a cyberwarfare weapon," IEEE Security & Privacy, 9(3), 49-51.

T. Halpin, Object-Role Modeling, http://www.orm.net.

T. M. Chen, J. C. Sanchez-Aarnoutse, & J. Buford (2011), Petri Net Modeling of Cyber-Physical Attacks on Smart Grid," IEEE Transactions on Smart Grid, 2(4), 741--749.

US-CERT, Control System Security Program: Overview of Cyber Vulnerabilities, http://www.us-cert.gov/control_systems/csvuls.html

VIKING project (2010), D2.3 SCADA system architecture.

V. M. Igure, S. A. Laughter, & R. D. Williams (2006), Security issues in SCADA networks, Computers & Security, 25(7), 498-506.