FP7-SEC-2011-1 Project 285647

Cybersecurity on SCADA: risk prediction, analysis and reaction tools for Critical Infrastructures

D3.1- Requirements and Reference Architecture of the Analysis and Detection Layer

General information	
Submission date	8/07/2013
Dissemination level	Confidential
State	Final Version
Work package	WP3000 - Cyber analysis and detection
Task	Task 3001 – Requirements and Reference Architecture of the Analysis and Detection Layer
Delivery date	30/06/2013

Editors

Name	Organisation
T. Cruz, P. Simões, J. Proença	University of Coimbra

Authors

Name	Organisation
T. Cruz, P. Simões, J. Proença, Pedro Alves, Luis Rosa, Jorge Barrigas, M. Curado, E. Monteiro,	University of Coimbra
E. Ciancamerla, A. Di Pietro, M. Minichino, S. Palmieri	ENEA
M. Ouedraogo, C. Feltus, D. Khadraoui	CRP Henry Tudor
A. Graziano	SELEX
M. Aubigny, C. Harpes	iTRUST
Lasith Yasakethu	SURREY
L. Lev	IEC
T. E. Roman	TRANS
A. Kvinnesland	LYSE

Reviewers

Name	Organisation	Date
D. Macone	CRAT	27/06/2013
C. Foglietta	ROMA3	25/06/2013

Executive Summary

This document provides an overview of the requirements and proposed reference architecture for the Analysis and Detection Layer of the CockpitCI solution for Industrial Control Systems (ICS) within Critical Infrastructures (CI), in line with the WP3000 task schedule.

ICS include supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), distribution management system (DMS-SCADA), energy management control system (EMS-SCADA) and other control system configurations such as Programmable Logic Controllers (PLC), normally used in industrial contexts such as utilities (electricity transmission and production, water supply and sewer processing, natural gas distribution), oil complexes or chemical processing, among others. SCADA systems control dispersed assets using centralized data acquisition and supervisory control, being used to control production systems such as factories or power plants, using supervisory and regulatory control. These control systems are vital to the operation of critical infrastructures that are often highly interconnected and mutually dependent systems.

Starting with an overview of SCADA architectures and related technologies, this document next delves into the subject of security, with an overview of the security record of SCADA technologies and the reasons for its current state. Related work and other initiatives regarding the security of critical infrastructures are also explored, before starting the definition and description of the reference architecture of the CockpitCI cyber detection and analysis layer, its requirements and fundamental components.

The subject of this document involves a vast array of concepts and knowledge from fields as diverse as systems management, control systems, communication networks, just to mention a few. Therefore, before going into detail on its main subject, there are several introductory chapters, which aim not only to provide enough background to the reader to understand the involved concepts, but also to make the document auto consistent.

Disclaimer: *All entities with access to this document in its present form, will not at any time or in any way, either directly or indirectly, use for personal benefit or divulge, disclose, or communicate any proprietary information hereby included, without prior consent of its intellectual property owners. This document must be protected and be treated as strictly confidential until further notice.*

Table of contents

List of figures

List of tables

1 Introduction

This document addresses the identification of relevant requirements for the analysis and detection layer of the CockpitCI platform, as well as the definition of a reference system architecture, which can be used as guideline for the subsequent research and design activities. Despite the fact that most activities in WP3000 are concentrated at the beginning of the work package, it will be kept running until near the end of WP3000, as an active reference point for the other activities of the work package.

1.1 Context

Industrial Control Systems (ICS) include supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC) are normally used in industrial contexts such as utilities (electricity distribution and production, water supply and sewer processing, natural gas distribution), oil complexes or chemical processing, among others.

Among ICS, SCADA systems are the largest group, being used to control dispersed assets using centralized data acquisition and supervisory control. These systems have evolved from proprietary and closed architectures to open, standards-based solutions, which are designed to ease interoperability with other similar platforms and different devices and platforms. However, this trend also had the drawback of increasing the security risks associated with these platforms, as a result of increased connectivity (with the Internet and other communication networks) and interoperability needs.

This is a result of the fact that vendors and even developers have failed to foresee the potential problems of exposing such systems, thanks to a generally accepted mindset, typical of the decade old SCADA platform paradigm, in which security was implicitly guaranteed by obscurity and systems isolation. The world of Industrial Control System for CI has proceeded mostly on its own path, lagging behind the advances in information technology and cyber-security practices. This is no more acceptable and there is the need to complement business awareness with cyber awareness to reach a superior level of awareness (global awareness).

Nowadays, ICS constitute a critical and strategic asset that is being increasingly targeted by malicious attacks, therefore increasing the potential for catastrophic consequences. In fact, in the last decade, these systems have been involved in a considerable number of incidents, of which Stuxnet is one of the best examples.

In this context, the CockpitCI project is an evolution of the MICIE project [MICIE], from which it inherited its core concept of increasing the cooperation among infrastructures to provide the operator with a better situation awareness in the presence of adverse events, with the purpose of increasing the CI level of service (business continuity). CockpitCI extended the MICIE philosophy, by encompassing awareness for cyber events, The CockpitCI aims to prove that the convergence among business continuity and cyber security is possible with a positive return for everyone involved, incorporating SCADA-oriented cyber awareness into the ICS infrastructure, using agents capable of detecting anomalies or intrusion attempts. These agents provide an ICS security feed that can be merged with the Information and Communication Technologies (ICT) network security feed, providing a broader insight into the security status of the whole infrastructure.

Moreover, a near real-time risk evaluation capability, which is built on the cyber-awareness mechanisms helps SCADA operators to better evaluate and react to potential threats, avoiding cascading effects, in line with existing service level agreements and availability levels contractually established with customers. As such, CockpitCI aims at reshaping the boundaries of the ICS and cyber-security contexts, in such a way that it becomes possible for both to work in tandem.

In the CockpitCI perspective, global awareness may transcend the local and even national level, as cyber issues are frequently equally unrestricted in terms of context and reach. This is possible by aggregating the information originated from the various control rooms of the infrastructure, from the control rooms of interdependent CIs, from the control rooms at national level that are connected with the intelligence at national and transnational level.

By encompassing both the local, system-specific perspective and the global view at the CI level, CockpitCI provides the means for a smarter and more effective reaction capability, targeting a graceful degradation scenario, thanks to deeper understanding of how much of the system can be kept in operation safely in adverse situations and maintaining at least partial operations rather than total shutdown.

1.2 Objectives

The CockpitCI system will incorporate several advanced real-time detection mechanisms, integrated on a cyber analysis and detection layer within the CI of the ICS. This detection layer is designed in such a way that it can integrate several different detection strategies, distributed along different levels, namely:

- Detection agents and field adaptors, including agents, adaptors and extensions for existing system components, as well as specialized network probes and honeypots to be added to the network which are able to capture behaviour or traffic patterns.
- A Dynamic Perimeter Intrusion Detection System, performing many of the tasks traditionally associated with a Distributed Intrusion Detection System (with the enhancements provided by the novel detection techniques). Accordingly with the CokcpitCI vision, it must be also be able to deploy prevention strategies of isolation.

To fulfil this aim, this architecture will have to accommodate different types of mechanisms, namely:

- Adaptive machine learning, including innovative data mining and pattern recognition approaches towards event correlation, innovations in dynamic Bayesian networks, artificial neural networks, vector machines, fuzzy and evolutionary systems.
- Aggressive usage of topology and system-specific detection mechanisms, based on the fact that the role and behaviour of each system component are expected to be more consistent over time than on other types of networks. The plan is to dynamically feed the intrusion and anomaly detection models with knowledge provided by a number of system specific sources, such as topology databases, policy databases, and trust-based mechanisms, as well as strategically placed honeypots.

This document will present the requirements and the reference architecture for the cyber-detection layer, with a particular focus on the innovative aspects of the proposition, but also detailing the integration, management, auditing and event aggregation mechanisms that constitute the whole layer.

1.3 Document structure

Given the broad scope of the concepts and technologies involved in design of the analysis and detection layer, this document includes an overview of related concepts and developments, for the sake of completeness and consistency.

The chapters of the document respectively deal with:

- Chapter 2 provides an introduction to ICS/SCADA systems, its architecture, components and technologies. It aims at provide a basic introduction to the world of SCADA technologies, using a simple example as the starting point.
- Chapter 3 provides an overview of the security problems of SCADA systems, their vulnerabilities, attack vectors and security incidents. It offers several insights into the specificities of ICS technologies, establishing comparisons with ICT contexts, whenever possible.
- Chapter 4 examines related work, both within European projects and cyber-security standards from several organizations and standardization entities.
- Chapter 5 is the heart of the document, presenting and discussing the reference cyber-analysis and detection layer architecture for the CockpitCI project. It also offers an overview of existing security mechanisms. Starting with an overview of existing security countermeasures, it proceeds with an analysis of detection mechanisms that can be used to find undisclosed vulnerabilities or detect attacks.
- Chapter 6 concludes this document, with a synthetic description and analysis of its contributions.

The text of document includes original statements and figures as extracted by the papers/web sites/documents/standards/guidelines considered along the process of knowledge acquisition. Only public documents have been considered, being properly referred along this document.

1.4 Intellectual property rights and dissemination level

The Dissemination Level of this deliverable is strictly "Confidential", in order not to interfere with ongoing Intellectual Property protection actions. Once these actions are successfully completed the dissemination level of this deliverable will be revised, in order to allow for wider dissemination of its content (or, at least, specific parts of it).

1.5 Acronyms and symbols

Terminology	Description
AAA	Authentication, Authorization, Accounting
A-Blocks	Analysis Boxes
AC	Application Control
ACL	Access Control List
ADU	Application Data Unit
AES	Advanced Encryption Standard
AFTER	A Framework for electrical power sysTems vulnerability identification, dEfense and Restoration
AGA	American Gas Association
AGC	Automatic Generation Control
ANSI	American National Standards Institute
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
API	Application Programming Interface
APT	Advanced Persistent Threats
ARP	Address Resolution Protocol
AS	Autonomous System
ASCII	American Standard Code for Information Interchange
ASDU	Application Service Data Unit
AVOIDIT	Attack Vector, Operational Impact, Defense, Information Impact, and Target
BDD	Bad Data Detection
BMS	Backup Master Station

BPDU	Bridge Protocol Data Unit
BSI	British Standard Institute
C&C	Command and Control
CA	Contingency Analysis
CC	Common Criteria
CC	Common Criteria
CERT	Computer Emergency Response Team
CGI	Common Gateway Interface
CI	Critical Infrastructures
CIA	Confidentiality, Integrity, and Availability
CIDF	Common Intrusion Detection Framework Architecture
CIIP	Critical Information Infrastructure Protection
CIP	Critical Infrastructure Security
COTS	Commercial Off The Shelf
CRC	Cyclic Redundancy Check
CRISALIS	CRitical Infrastructure Security AnaLysIS
CVE	Common Vulnerabilities and Exposures
CySeMoL	Cyber Security Modeling Language
DAI	Dynamic ARP Inspection
DARPA	Defense Advanced Research Projects Agency
D-Blocks	Database Blocks
DBMS	Database Management Systems
DCS	Distributed Control System
DDoS	Distributed Denial of Service
DFC	Data Flow Control Bit
DHCP	Dynamic Host Configuration Protocol
DHS	Department of Homeland Security
DIDS	Distributed IDS
DIR	Direction Bit
DLL	Dynamic Link Library
DM	Derived Measure
DMS	Distribution Management Systems
DMZ	Demilitarized Zone

DNP3	Distributed Network Protocol
DNS	Domain Name System
DNSSEC	DNS Security
DoS	Denial of Service
DoS	Denial of Service
DoW	Description of Work
EAAT	Enterprise Architecture Analysis Tool
EAL	Evaluation Assurance Levels
EAL	Evaluation Assurance Level
E-Boxes	Event Boxes
EIA	Electronic Industries Association
EMS	Energy Management System
ENEA	Italian National Agency for New Technologies, Energy and Sustainable Economic Development
ENEL	Ente Nazionale per l'energia ELettrica
EPA	Enhanced Performance Architecture
ESCoRTS	European network for the Security of Control and Real-Time Systems
ESP	Electronic Security Perimeter
EU	European Union
FC	Function Code
FCB	Frame Count Bits
FCV	Frame Count Valid bit
FIFO	First In, First Out
FN	Function Name
FP7	Seventh Framework Programme
FSM	Field Security Manager
FTP	File Transfer Protocol
GDP	Gross Domestic Product
HIDS	Host Intrusion Detection Systems
HMI	Human-Machine Interaction
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol over Secure Socket Layer

ICCP	Inter-center Control Communications Protocol
ICMP	Internet Control Message Protocol
ICS	Industrial Control Systems
ICT	Information and Communication Technologies
IDC	Internet Data Center
IDMEF	Intrusion Detection Message Exchange Format
IDPS	Intrusion Detection & Prevention System
IDS	Intrusion Detection System
IDWG	Intrusion Detection Working Group
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Devices
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
INSPIRE	Increasing Security and Protection through Infrastructure REsilience
IP	Internet Protocol
IPSec	Internet Protocol Security
IRP	Integrated Risk Prediction
ISA	International Society of Automation
ISO/IEC	International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC)
ISP	Internet Service Provider
IT	Information Technology
ITU-T	International Telecommunication Union-Telecommunication Standardization Sector
LAN	Local Area Network
LanMan	LAN Manager
LBNL	Lawrence Berkeley National Laboratory
LCCI	Large Complex Critical Infrastructures
LML	Log Management Lackey
LOIC	Low Orbit Ion Cannon
MAC Address	Machine Access Control Address
MBAP	Modbus Application Header
MIS	Management Information Systems

MITM	Man-In-The Middle
MTU	Maximum Transfer Unit
NASL	Nessus Attack Scripting Language
ND	Night Dragon
NERC	North American Electric Reliability Corporation
NIDS	Network Intrusion Detection Systems
NIST	National Institute of Standards and Technology
NIST	National Institute of Standards and Technology
OCSVM	One Class Support Vector Machines
OLE	Object Linking and Embedding
OOB	Out of Band
OPC	OLE for Process Control
OS	Operating Systems
OSI	Open Systems Interconnect
PC	Personal computer
PCA	Principal Components Analysis
PDU	Protocol Data Unit
PIDS	Perimeter Intrusion Detection System
PLC	Programmable Logic Controller
PP	Protection Profile
PRECYSE	Prevention, protection and REaction to CYber attackS to critical infrastructures
PRM	Primary Bit
QoS	Quality of Service
RAT	Remote Access Trojan
RBAC	Role-Based Access Control
R-Blocks	Reactive Blocks
RDBMS	Relational Database Management System
RES	Reserved
RFC	Request for Comments
RTOS	Real-Time Operating System
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition

SE	State Estimator
SEC	Simple Event Correlator
SIR	Susceptible, Infected and Removed
SMGW	Secure Mediation Gateway
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SRC	Security Relevant Components
SSL	Secure Sockets Layer
ST	Security Targets
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
TLC	Telecommunication
TSO	Transmission System Operator
UART	Universal Asynchronous Receiver/Transmitters
UDP	User Datagram Protocol
US	United States
USB	Universal Serial Bus
VIKING	Vital Infrastructure, Networks, Information and Control Systems Management
VLAN	Virtual Local Area Network
VoIP	Voice over IP
VPN	Virtual Private Networking
WAN	Wide Area Network
WMI	Windows Management Instrumentation
WWW	World Wide Web

2 An Overview of SCADA Architectures

2.1 Industrial Control Systems within a CI

Industrial Control Systems (ICS) are command and control networks and systems designed to support industrial processes. These systems are responsible for monitoring and controlling a variety of processes and operations such as gas and electricity distribution, water treatment, oil refining, railway transportation, etc.

The largest subgroup of ICS is SCADA (Supervisory Control and Data Acquisition) systems. In the last few years, ICS have passed through a significant transformation from proprietary, isolated systems to open architectures and standard technologies highly interconnected with other corporate networks and the Internet. Today, ICS products are mostly based on standard embedded systems platforms, applied in various devices, such as routers or cable modems, and they often use commercial off-the-shelf software. All this has led to cost reductions, ease of use and enabled the remote control and monitoring from various locations. However, an important drawback derived from the connection to intranets and open communication networks, is the increased vulnerability to computer network-based attacks.

Figure 2-1 shows a hierarchy of logical levels, proposed by ISA99 [ISA99] series of standards for characterize the Industrial Control System (ICS) of a generic integrated manufacturing or production system, which well characterize also the ICS of a Critical Infrastructure.

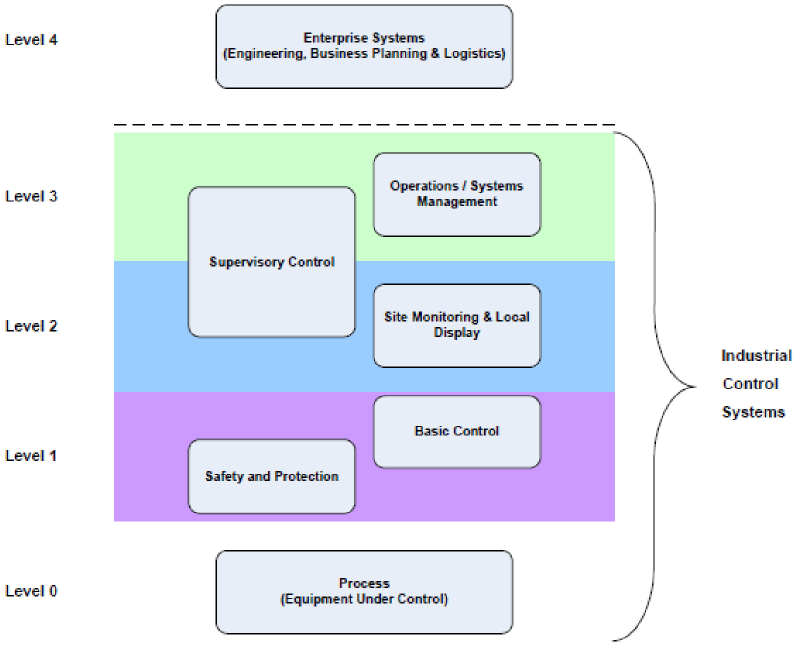


Figure 2-1: ICS within production system hierarchy [ISA99]

The range of logical levels is from level 0 to level 4 and in such a hierarchy ICS components are spread among levels 1, 2 and 3. Level 0 is the lower bound of ICS and includes the sensors and actuators directly connected to the process and process equipment.

Level 4, **Enterprise Systems**, is defined as including the functions involved in the business-related activities needed to manage a manufacturing organization. Functions include enterprise or regional financial systems and other enterprise infrastructure components such as production scheduling, operational management, and maintenance management for an individual plant or site in an enterprise. For the purposes of this standard, engineering systems are also considered to be in this level.

Level 3, **Operations Management**, includes the functions involved in managing the work flows to produce the desired end products. Examples include dispatching production, detailed production scheduling, reliability assurance, and site-wide control optimization.

Level 2, **Supervisory Control**, includes the functions involved in monitoring and controlling the physical process. There are typically multiple production areas in a plant such as distillation, conversion, blending in a refinery or the turbine deck, and coal processing facilities in a utility power plant. Level 2 functions include:

- Operator human-machine interface.
- Operator alarms and alerts.
- Supervisory control functions.
- Process history collection.

Level 1, **Local or Basic Control**, includes the functions involved in sensing and manipulating the physical process. Process monitoring equipment reads data from sensors, executes algorithms if necessary, and maintains process history. Examples of process monitoring systems include tank gauging systems, continuous emission monitors, rotating equipment monitoring systems, and temperature indicating systems. Process control equipment is similar. It reads data from sensors, executes a control algorithm, and sends an output to a final element (e.g., control valves or damper drives). Level 1 controllers are directly connected to the sensors and actuators of the process. Level 1 includes continuous control, sequence control, batch control, and discrete control. Many modern controllers include all types of control in a single device. Also included in Level 1 are safety and protection systems that monitor the process and automatically return the process to a safe state if it exceeds safe limits. This category also includes systems that monitor the process and alert an operator of impending unsafe conditions. Safety and protection systems have traditionally been implemented using physically separate controllers, but more recently it has become possible to implement them using a method known as logical separation within a common infrastructure. Level 1 equipment includes, but is not limited to:

- DCS controllers

- PLCs
- Remote Terminal Units (RTU).
- Meters
- Other field embedded devices

Safety and protection systems often have additional safety requirements that may not be consistent or relevant to cyber security requirements. These systems include the safety systems in use in chemical and petrochemical plants as identified in the ANSI/ISA-84 series of standards, nuclear plant safety or safety-related systems as identified in the ANSI/ISA-67 series, and protective functions as identified in the Institute of Electrical and Electronics Engineers (IEEE) Power Engineering Society standards.

Level 0, **Process**, is the actual physical process. The process includes a number of different types of production facilities in all sectors including, but not limited to, discrete parts manufacturing, hydrocarbon processing, product distribution, pharmaceuticals, pulp and paper, and electric power. Level 0 includes the sensors and actuators directly connected to the process and process equipment.

This hierarchy will provide the basis for the cyber-detection detection architecture introduced in chapter 5, which attempts to separate security perimeters accordingly to their contextual position relatively to this classification.

2.2 Comparing ICS and ICT systems

With reference to Figure 2-1 the level 4, enterprise system relies on ICT technologies. For Critical Infrastructures, this level is constituted by the enterprise/corporate network.

Therefore, it makes sense to understand which are the main differences between Industrial Control System (ICS)/SCADA and Information and Communication Technology (ICT) systems within enterprise/corporate network, from an ICS security standpoint. Accordingly to [NIST], these can be synthesized as such:

Performance. ICS systems are *hard real-time* systems because of the need of completing an operation within a strict deadline in order not cause potential loss in safety, such as damaging the surroundings or threatening human lives. Timeliness expresses the time-criticality of control systems as it includes both the responsiveness aspect of the system, e.g. a command from controller to actuator should be executed in real-time by the latter, and the timeliness of any related data being delivered in its designated time period. Or in a more general sense, this property describes that any queried, reported, issued and disseminated information shall not be stale but corresponding to the real-time and the system is able and sensitive enough to process request, which may be of normal or of legitimate human intervention in a timely fashion, such as within a sampling period. In addition, the order of data arrival at central monitor room may play an important factor in the representation of process dynamics and affect the correct decision making of either the controlling algorithms or the supervising human operators. In contrast, ICT systems that deliver services like live audio-video are soft real-time systems as they may tolerate certain latency and respond with decreased service quality, (e.g., dropping frames while displaying a video). High throughput is typically not essential to ICS. Some ICS systems require deterministic responses.

Availability means that any component of a ICS system (may it be a sensory or servo mechanical device, communication or networking equipment, or radio channel, computation resource and information such as sensor readings and controller commands that transmits or resides within the system should be ready for use when is needed. Many ICS processes are continuous in nature. Unexpected outages of systems that control industrial processes are not acceptable. Outages often must be planned and scheduled days/weeks in advance. Exhaustive pre-deployment testing is essential to ensure high availability for the ICS. In addition to unexpected outages, many control systems cannot be easily stopped and started without affecting production. The use of typical ICT strategies such as rebooting a component, are usually not acceptable solutions due to the adverse impact on the requirements for high availability, reliability and maintainability of the ICS. Some ICS employ redundant components, often running in parallel, to provide continuity when primary components are unavailable**.**

Integrity requires data generated, transmitted, displayed and stored within ICS being genuine and intact without unauthorized intervention, including both its content, which may also include the header for its source, destination and time information besides the payload itself. A much related terminology is authenticity, in the content of ICS, it implies that the identity of sender and receiver of any information shall be genuine. By using this definition of integrity, then authenticity falls within the same category.

Confidentiality refers to that unauthorized person should not have any access to information related to the specific ICS. At current stage, this need is dwarfed by the desirability of availability in a control performance centric setting. ICS systems measure and control physical processes that generally are of a continuous nature with commands and responses are simple and repetitive. Thus the messages in ICS are relatively easy to predict. Hence confidentiality is secondary in importance to data integrity. However, the confidentiality of critical information such as passwords, encryption keys, detailed system layout map and etc. shall rank high when it comes to security concerns in industry.

Risk Management. In a typical ICT system, data confidentiality and integrity are typically the primary concerns. For an ICS, human safety and fault tolerance to prevent loss of life or endangerment of public health or confidence, regulatory compliance, loss of equipment, loss of intellectual property, or lost or damaged products are the primary concerns. The personnel responsible for operating, securing, and maintaining ICS must understand the important link between safety and security.

Physical Interaction. In a typical ICT system, there is not physical interaction with the environment. ICS can have very complex interactions with physical processes and consequences in the ICS domain that can manifest in physical events. All security functions integrated into the ICS must be tested (e.g., off-line on a comparable ICS) to prove that they do not compromise normal ICS functionality.

Time-Critical Responses. In a typical ICT system, access control can be implemented without significant regard for data flow. For some ICS, automated response time or system response to human interaction is very critical. For example, requiring password authentication and authorization on an HMI (Human-Machine Interaction) must not hamper or interfere with emergency actions for ICS. Information flow must not be interrupted or compromised. Access to these systems should be restricted by rigorous physical security controls. The use of encryption could require some tasks to be performed by ICS and the processes within each task could require to be interrupted and restarted. The timing aspect and task interrupts can preclude the use of conventional encryption block algorithms that instead are broadly used in ICT for applications like e-commerce or financial applications.

System Operation. ICS operating systems (OS) and applications may not tolerate typical ICT security practices. Legacy systems are especially vulnerable to resource unavailability and timing disruptions. Control networks are often more complex and require a different level of expertise (e.g., control networks are typically managed by control engineers, not ICT personnel). Software and hardware are more difficult to upgrade in an operational control system network. Many systems may not have desired features including encryption capabilities, error logging, and password protection.

Resource Constraints. ICS and their real time OSs are often resource-constrained systems that usually do not include typical ICT security capabilities. There may not be computing resources available on ICS components to retrofit these systems with current security capabilities. Additionally, in some instances, third-party security solutions are not allowed due to ICS vendor license and service agreements, and loss of service support can occur if third party applications are installed without vendor acknowledgement or approval.

Communications. Communication protocols and media used by ICS environments for field device control and intra-processor communication are typically different from the generic ICT environment, and may be proprietary.

Change Management. Unpatched software represents one of the greatest vulnerabilities to a system. Software updates on ICT systems, including security patches, are typically applied in a timely fashion based on appropriate security policy and procedures. In addition, these procedures are often automated using server-based tools. Software updates on ICS cannot always be implemented on a timely basis because these updates need to be thoroughly tested by the vendor of the industrial control application and the end user of the application before being implemented and ICS outages often must be planned and scheduled days/weeks in advance. The ICS may also require revalidation as part of the update process. Another issue is that many ICS utilize older versions of operating systems that are no longer supported by the vendor. Consequently, available patches may not be applicable.

Component Lifetime. Typical ICT components have a lifetime on the order of 3 to 5 years, with brevity due to the quick evolution of technology. For ICS where technology has been developed in many cases for very specific use and implementation, the lifetime of the deployed technology is often in the order of 15 to 20 years and sometimes longer.

Access to Components. Typical ICT components are usually local and easy to access, while ICS components can be isolated, remote, and require extensive physical effort to gain access to them.

Graceful degradation requires the system being capable of keeping the attack impact local and withholding data flow that may escalate into a full cascading event.

Memory allocation. In ICS systems memory allocation is usually more critical than in conventional ICT systems because many field level devices in ICS system are embedded systems that run years without rebooting but accumulating fragmentation with the consequence of a program stall.

2.3 Evolution of ICS systems

The origin of (ICS)/SCADA systems goes back to the 1960's (in the mainframe era), having been around as long as there have been control systems. Prior to SCADA, early systems were very rudimentary – data acquisition was performed by means of panels of meters, lights and strip chart recorders. The operator manually operating various control knobs exercised supervisory control. These devices were and still are used to do supervisory control and data acquisition on plants, factories and power generating facilities.

These original systems were very simple by nature, mainly because there were no formal data processing or memory mechanisms involved, being little more than reactive systems interconnecting sensors and visual indicators. However, that simplicity was also their main drawback, being unfeasible for usage anything other than small-scale and physically limited scenarios. Also since data logging was impossible, error or failure debugging capabilities was very limited – this also impacted the ability to perform long-term performance monitoring.

With time, ICS systems evolved to their present situation, with the use of DCS (Distributed Control Systems), PLCs (Programmable Logic Controllers) and more evolved data processing and networking technologies. However, some of their components (for instance, PLCs) have a lifecycle that frequently spans several decades (it is common to find devices based or architectures with 20 or more years in use). This longevity has to do with maturity – a feature favoured in critical systems, traditionally associated with reliability.

One of areas where ICS systems have evolved considerably is in terms of their communication and interoperability capabilities. While original systems were isolated and self-contained by nature, they progressively started to open to the exterior world, making use of data communication networks for its own internal purposes and to share information with the outside world or even other systems. These connections might exist for various reasons: with the ICT corporate Local Area Network (LAN), to exchange information with performance auditing or stock management applications; a Wide Area Network (WAN) connection to connect to other facilities (for instance, two power stations) or to an operations control center, separated miles away. Such WAN connections might be ensured using leased lines, dial-up or, more recently the Internet itself [Ten2008] [Davis2006] – also, it is frequent for original device manufactures to provide remote assistance using such mechanisms.

Also, proprietary equipment and protocols were also the norm on older (ICS)/SCADA systems, limiting interoperability between devices from different manufacturers (and sometimes, albeit less frequently, between different models of the same manufacturer). This created a situation of vendor lock-in that forced the customer to remain attached to a specific device family form a particular manufacturer due to the cost of migration. Presently, equipment and protocols have been standardized, with the adoption of COTS (*Commercial Off-The-Shelf)*, equipment whether possible (for instance for LAN communication) [Igure2006].

As a consequence of the introduction of data processing capabilities to SCADA systems, together with the evolution of embedded systems, operating systems also became part of the ICS ecosystem that evolved with time. From proprietary systems, the situation evolved up to the point where Windows or Unix-derivatives [Creery2005] are being used, together with real-time operating systems such as VXWorks or Real-Time Linux [Davis2006].

This evolution brought significant benefit to ICS systems, in terms of functionality, rationality and cost. However, they are also closely related with some of the most important security issues that CockpitCI tries to address, as shown in subsequent chapters.

2.4 Architecture of ICS systems

This section presents the general architecture of a SCADA system, including its components and the communication flows between them, using a simple example to explain its operation and fundamental building blocks. This will be followed by a more thorough discussion of each specific component.

2.4.1 A typical SCADA system

In generic terms, a SCADA system includes consists of a number of remote terminal units (or RTUs) collecting field data connected back to a master station via a communications system. The master station displays the acquired data and also allows the operator to perform remote control tasks. Accurate and timely data (normally real-time) allows for optimization of the operation of the ICS, with the benefit of a more efficient, reliable and most importantly, safer operation.

On a more complex SCADA system there are essentially five levels or hierarchies [Bailey2003]:

- Master station(s)

- Slaves/RTUs
- Field level instrumentation and control devices
- Communication networks
- ICT level

Figure 2-2 represents a simple SCADA system, including some of the main components. This system encompasses a Master Station, Slaves, Field Network and Field components (sensors and actuators).

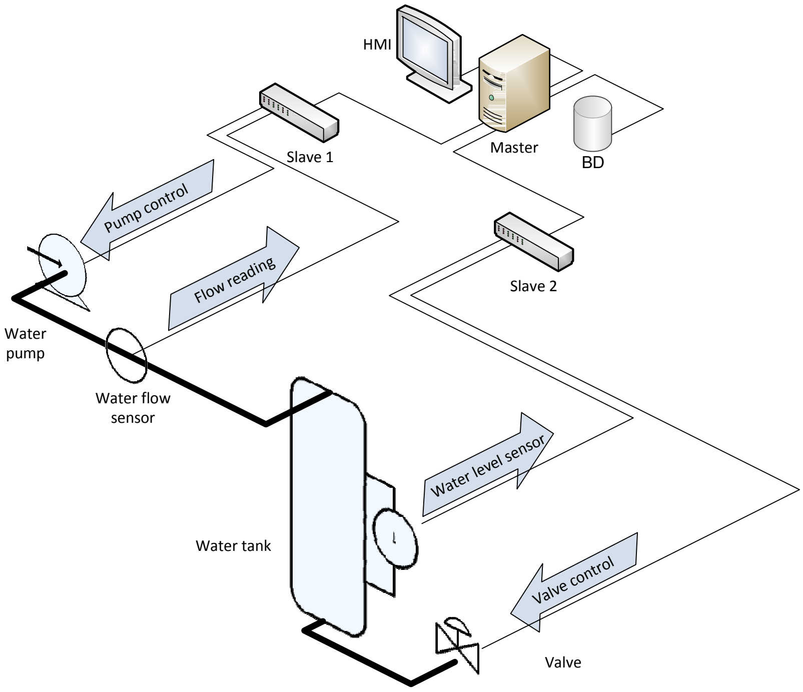


Figure 2-2: Architecture for a simple SCADA system. (Adapted from [Bailey2003])

Sensors 1 and 2 provide information about the water flow in a pipe and the water level in the tank. As for actuators, we have a water pump and a valve, which controls the water output from the tank. The operation of this system can be described in a step-by-step basis:

- When initialized the Master Station requests reading from the Slaves to obtain information about water flow debit from the pump and water level on the tank.
- Accordingly with those values, the Master Station defines which ones are the desired parameters for the sensors, which are communicated to the Slaves.
- The two slaves are able to autonomously control the actuators to maintain the parameters within defined values. For instance, Slave 1 is able to adjust the pump rotation accordingly to maintain flow reading within predefined levels. Slave 2 performs similarly, by opening and closing the valve accordingly with the water level.

- In runtime, the Master Station is able to obtain periodic readings from the Slaves, being able to maintain a log of the global state or operational parameters of the system. These data is stored in one or more databases, separated by context – one for data from the controlled process and another one for historic log [Björkman2010].

For a better explanation of the involved concepts, the next subsections will delve into each component with more detail.

2.4.1.1 Master stations

Master Stations are critical system components, controlling and monitoring Slaves when needed. Master Stations also provide interfaces for HMI (Human-Machine Interaction) display consoles for display of information and control of the remote site, use for interfacing with human operators, which are the main responsible for the correct operation. HMIs enable one or more operators to take care of the system, monitoring and controlling its operation – for instance, reporting alerts and enabling operators to manually send commands to Slaves, if needed.

Master Stations are frequently connected to other applications, as it is the case for databases. In this situation, several different databases might be involved: for instance a Relational Database Management System (RDBMS) for logging historic data from all operation parameters (sensor data, for instance) or a real-time Database Management Systems (DBMS) [Björkman2010] for the continuous update of the system state. The use of real-time DBMS for the latter case has to do with the need to update information in a real-time, continuous basis, with a minimal latency between contextual changes in the process state and its consequent update in the database [VIKING2010d] – this kind of performance is not possible with conventional RDBMS [Björkman2010].

There are normally three components on a Master Station: the operating system software; the system SCADA software (suitably configured) and the SCADA application software. n modern SCADA systems, Master Stations are hosted by standard PCs with standard operating systems, such as Windows or Unix-derivatives [Krutz2006]. Also, several Master Stations might exist, depending on the process topology or adopted communication technologies protocol (for instance, some protocols do not allow for multi-master operation).

2.4.1.2 Slaves

This equipment is connected to one or more Master Stations, and also to sensors and actuators. It is responsible for the majority of the monitoring and control activities – being an embedded system; it normally has a limited computation capability. Slaves/RTUs provide an interface to the field analogue and digital signals situated at each remote site

Slaves also receive messages from Master Stations. These messages can be requests for readings of sensor values, commands for actuator components or program downloads. Slaves can also have responsibilities in terms of process control, being able to implement a control loop for adjusting actuator properties accordingly with sensor outputs – in such situations, PLCs are frequently used to implement smart RTUs.

Slaves receive information from process sensors and send it to the Master Station, normally in response to information requests. Also, Slaves may perform control actions, either autonomously (in the case of a PLC), with the Master Station establishing its basic operational parameters or being directly controlled by the Master Station. Nevertheless, Slaves do not have complete knowledge of the controlled processes, being restricted to a limited scope.

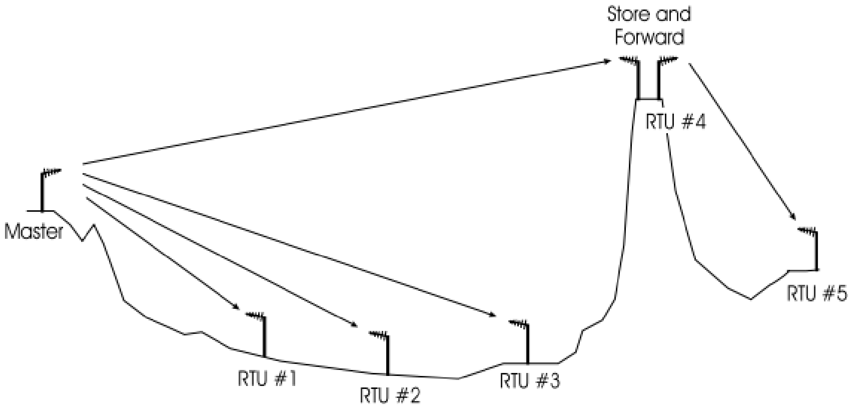


Figure 2-3: Slave with *Store And Forward* operation [Bailey2003]

Another kind of functionality performed by Slaves has to do with Store and Forward capabilities. When an RTU is unable to communicate directly with a Master Station (for instance, in a wireless radio network), an intermediate unit may be used as a message repeater (see Figure 2-3).

2.4.1.3 Sensors and actuators

Field devices (sensors and actuators) provide the interface with the physical world, providing information about the process and enabling the execution of actions that affect its behaviour. These elements are directly connected to the controlled process: sensors are able to obtain information about physical states or properties (such as temperatures, fluid levels, pressure, conveyor belt speeds, among others), detecting any variation in these properties.

Actuators provide the ability for controlling the process behaviour at the physical level, like an electrovalve, an electrohydraulic actuator, a servo controller, among others.

2.4.1.4 Communication network technologies

When it comes to communication networks, modern SCADA systems incorporate the DCS (Distributed Control System) paradigm, incorporating at least two different networks:

- **Control Network:** it contains all slaves (RTUs and/or PLCs). It directly interfaces with the field network, i.e. the network of actuators and sensors that physically perform the process tasks on the system. It is also connected to the Process Network.
- **Process Network:** composed by the Master Stations and all the other systems that gather the data coming from the Control Network and send commands to RTUs and PLCs though the Control Network.

Control Networks provide the pathway for communications between the Master Stations and the Slaves. Starting with the first proprietary implementations were proprietary [Igure2006], the 1980's and 90's saw the rise of standardized point-to-point or bus-based serial communication topologies such as the Electronic Industries Association (EIA) standards, EIA-232 or EIA-485 [IDC2009] differential data transmission.

SCADA communications over a serial channel require each device on the same channel to use the same settings so that master device can communicate with each slave device on the channel. The following parameters define how a serial channel functions:

- Communications channel
- Baud rate (typically: 300, 1200, 2400, 4800, 9600, 19,200)
- Number of bits (8 or 7)
- Parity (none, even or odd)
- Number of stop bits (typically 1)
- Flow control and error-checking (CRC, none, XON/XOFF)
  (Note: Flow control is rarely used in SCADA communications)

All devices on a RS-232, RS-422, or RS-485 must be configured with the same communications settings. Serial communications systems are used over various wired, fibre, or wireless media networking technologies, namely:

- **Serial communication over telephone carrier lines** using plain old telephone service (POTS) modem lines. This was a common method for SCADA systems to connect the centralised control room equipment to the field components. In such cases, modems connected master SCADA PCs to remote field devices over RS-232 serial communications circuit.
- **Serial communications over TDM (Time Division Multiplexed) circuits are** used in SCADA, Energy Management Systems and Distributed Control Systems to connect the RTU, controllers and other equipment. TDM circuits support a multiplexed hierarchy that is able to provide either large bandwidth circuits like T1 and DS3, specialized proprietary trunks for CCTV, or smaller individual RS-232 circuits for polling PLCs and RTUs.
- **Serial communications over wireless RF systems** are designed and deployed to provide communications coverage between devices in the field and the control room

systems. These wireless systems are supported with either licensed radios that use known fixed transmit and receive frequencies, or with spread spectrum radios that use frequency hopping techniques to avoid having to use fixed RF frequencies. Both licensed and non-licensed wireless RF systems were initially only designed to transport serial communications.

Serial protocols only allow for one master to communicate to one slave at a time, and while the slave is responding to the master, it is consuming a serial channel that cannot be used for any other purpose. This may difficult remote troubleshooting and management of field devices using serial communications, up to the point that is frequent having modifications and device management to be performed out in the field.

More recently, the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol family has become the norm in SCADA environments. As IP technologies became increasingly popular on SCADA systems, the last components to migrate were embedded devices, such as field controllers, meters, instrumentation, and related telecommunications systems linking the control room computing systems with field devices.

Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and even Smart Instrumentation began to ship with Ethernet communications in the late 1990s up to the point that, nowadays, nearly every control system vendor offers their equipment with Ethernet TCP/IP communications capability. As most SCADA and ICS vendors acquired TCP/IP stacks and drivers from other third parties and integrated them into their hardware, they also needed a TCP/IP-based protocol for the computer systems in the control room to communicate with the hardware. To solve this problem, a number of vendors decided to simply encapsulate their serial protocols with TCP/IP headers or wrappers, and re-use the same underlying protocols – most of which do not support authentication or encryption and which transmit data in clear text.

Therefore, some serial-era protocols (such as Modbus) were adapted for operation in TCP/IP networking environments, with the benefit of lower cost (by using COTS equipment and technologies) and increased reliability and availability (for instance, the creation of redundant infrastructures can be eased by adopting Ethernet-based TCP/IP topologies), yet at the expense of sacrificing security. The same rationale also applies to those cases where operators enabled serial communications over TCP/IP networks by using serial to IP adaptors (like RS-232/Ethernet TCP/IP adaptors) at each endpoint (for instance, between SCADA servers and legacy devices).

2.4.1.5 Communication protocols for SCADA systems

SCADA communication protocols are one of the most critical parts of ICS system operations, being responsible for retrieving information from field equipment and for sending control commands. Recent estimates revealed the existence of between 150 and 200 different protocols for this purpose, the majority of them being proprietary [Igure2006]. This section deals with the operation of two of the most popular protocols [ESCoRTS2010c] [Igure2006] being used: Modbus and DNP3. Also, an extension of the International Electrotechnical

Commission (IEC) 60870-5-101 standard is described, which adds support for TCP/IP network, also a good candidate along with the mentioned protocols.

Modbus

The Modbus protocol, originally developed by Modicon (currently part of the Schneider Electric Group) in 1979, is one of the most popular protocols being used for SCADA applications, mainly thanks to its simplicity and ease of use. In fact, recent survey in the American Control Engineering magazine indicated that over 40% of industrial communication applications use the Modbus protocol.

There are several variants of Modbus, with Modbus RTU and Modbus TCP being the most popular – there is also a Modbus American Standard Code for Information Interchange (ASCII) (basically Modbus RTU with text-readable messages for easier debugging – not used in production environments for efficiency reasons [Pauli2003]), Modbus User Datagram Protocol (UDP) (like Modbus TCP, but using UDP for reduced overhead) or Modbus Plus (proprietary version from Schneider Electric, with extra functionality). This section will deal mostly with Modbus RTU and Modbus TCP, mostly because of their popularity.

Modbus RTU

Modbus RTU was originally designed for use with serial communication technologies, as it is the case with EIA-485. This is an application-level layer protocol, which is based in a pooling mechanism – only Master Stations can initiate communication. Each Master Station has knowledge of all its Slaves, broadcasting pooling requests sequentially, with each Slave/recipient responding accordingly. It is also possible to broadcast commands to all Slaves simultaneously, using the address “0” for this purpose.

This protocol limits the number of Master Stations to 1, with a theoretical limit of 247 Slaves per each Master (this limitation is due to the number of bit used for addressing). For practical reasons, the maximum number of Slaves is much lower, mainly because of performance limitations [IDC2009].

Each Slave has 4 different internal arrays: two for coils and two for registers. For each kind there is a different table for reading and writing. Coils are 1-bit values, used for binary sensors (a two-position switch, for instance), while registers are 16-bit long and can store more complex data types (such as continuous values).

Modbus RTU defines a simple protocol data unit (PDU – see Figure 2-4) that is independent from the underlying communication layers. To map Modbus on a specific bus or network a set of additional fields are added to the application data unit (ADU).

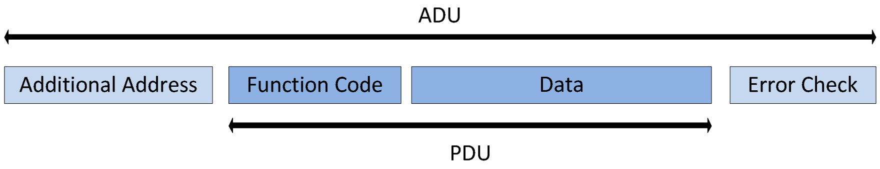


Figure 2-4: Modbus frame [Modbus2006])

The first field (Additional Address) contains the Slave address to which the message is destined. In the case of a response ADU, this field remains unchanged, in order to enable the Master Station to identify the sender - this makes sense because this protocol does not intend that slaves should be able to communicate with each other.

The function field (1 byte) indicates to the Slave the type of operation to execute. The data field is filled by the device or equipment that initiates a Modbus transaction and contains additional information (such as parameters – register addresses, for instance) that the Slave needs to execute the defined function action. In some situations the data field might not exist at all.

If the requested action is performed without problems, the data field on a response ADU contains the requested data – otherwise, the returning ADU will contain the exception error code. For a normal response, the returned function code repeats the original function - for an exception response, the Slave will return a code that is equivalent to the original function code specified in the received PDU with its most significant bit set to 1.

The last field is the error check field, containing a CRC-16 (Cyclic Redundancy Check), used for verification of message integrity.

Modbus TCP

Modbus TCP wad developed in order to add support for TCP/IP networks. Its operation philosophy is very similar to the Modbus RTU variant, with some changes in terms of message structure. Modbus TCP uses TCP ports 502 for Slaves and non-privileged ports (above 1024) for Masters. It is possible to convert Modbus RTU equipment for TCP operation, by using special-purpose gateways for translating between both variants [IDC2009].

The CRC16 error check field used in Modbus RTU is discarded in Modbus TCP because it is assumed that the TCP/IP protocol stack already offers integrity control for payloads.

Modbus TCP framing is based on RTU framing, with an identical PDU, without the remaining ADU fields (the Additional Address and Error Check fields are not used). A MBAP (Modbus Application Header) is added, containing the following fields:

- A transaction identifier (2 bytes), identifying the request to ensure coherence in case responses arrive out of sequence (in Modbus TCP, a slave can handle several requests simultaneously).

- A protocol identifier (2 bytes), filled by the Master station and always 0x0000.
- The length field contains the number of bytes used by the following fields.
- The last field, Unit Identifier is used when RTU devices are used in a TCP environment, through protocol gateways.

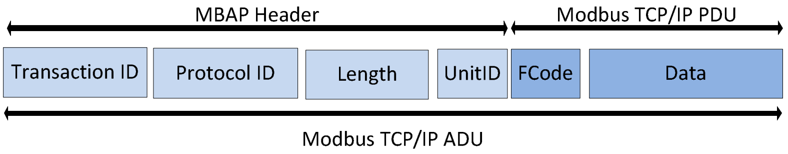


Figure 2-5: Modbus/TCP framing [Modbus2006])

The remaining fields, referent to the Modbus TCP/IP protocol PDU, relate to executed operations, as described next: Function code: This field can contain any integer value from 0 to 255, being the range of [0, 127] referent to requests sent by the Master and the range of [128, 255] related to responses sent by the Slave. The meaning of these values is explained with a simple pseudo-code in Figure 2-6, as follows:

Function Code Operation (pseudo-code)

```
START
IF (values between [128, 255])
        There was an error executing the command
ELSE IF (response value = request value)
        Slave executed the command
ELSE
        Slave alerts Master of an error
END
```

Figure 2-6: Function Code Operation

Note that the value 0 (zero) is never used, although at first it was referred that the range of these is [0, 255]. Reading the code above, if the values are between 128 and 255 it means that an error occurred and, in the other hand, if the values for the request and response match it means that the Slave ran the command successfully. In case none of the above conditions are met, the Slave notifies the Master that an error occurred.

Data bytes

The explanation for this field on the form of pseudo-code can be seen in Figure 2-7:

Data Bytes Operation (pseudo-code)

```
START
```

```
IF (execution = SUCCESS)
        The message contains the data requested by the Master
ELSE
        The message contains information about the error
END
```

Figure 2-7: Data Bytes Operation

In this scenario, in case the request execution has occurred successfully, it means that the message contained the (correct) information requested by the Master; otherwise this would only include information about the occurred error.

The functions used for reading, writing and other operations are categorized by class so these can be grouped into various types of transactions, as follows in Table 2-1[1]. This table only refers the function names (FN), code (FC) and respective classes.

Table 2-1: Transaction types

Class	Function Name (FN)	Function Code (FC)
Class 0	read multiple registers	3
	write multiple registers	16
Class 1	read coils	1
	read input discrete	2
	read input registers	4
	write coil	5
	write single register	6
	read exception status	7
Class 2	force multiple coils	15
	read general reference	20
	write general reference	21
	mask write register	22
	read/write registers	23
	read First In, First Out (FIFO) queue	24

The more primitive reading and writing operations referred in the previous table lie essentially in classes 0 and 1.

[1] There is a detailed table with all the functions used by the Modbus protocol in [Modicon1996]. Table 2-1 only intends to categorize these functions.

DNP3

DNP3 is a protocol originally developed in 1990 by Westronic (now GE-Harris Energy Control Systems), later transferred to the DNP Users Group, in 1993. In 2010 it became part as IEEE standard [IEEE1815-2010]. This protocol was developed with the electric industry in mind, but it is a generic protocol in such a way that in can be used in other types of industrial environments.

It is a more capable protocol, when compared with Modbus, but also more complex. It was initially developed for serial communications (e.g., EIA-485), but it was also ported to TCP/IP networks. DNP3 was designed to reduce the communications overhead and increase efficiency. Instead of full-state polling (like Modbus), DNP3 supports a mechanism called Event Data Reporting, which allows Slaves to store changes in sensor information, only reporting significant data changes.

Moreover, a Slave can be configured to perform event-oriented reporting, without Master pooling. DNP3 allows outstations to report data to one or more master stations using unsolicited responses (report by exception) for event data objects. The outstation reports data based about the assigned class of the data. For example the outstation can be configured to only report high priority class 1 data.

Data is stored in buffers, associated with Classes (numbered from 0 to 3). Class 0 is static and does no store events – a pool that stores immutable data. Classes 1 to 3 are associated with different priorities. Assuming class 1 contains the highest priority change event data and class 3 contains the lowest priority change event data, a class 1 poll would ideally be performed as often as possible, a class 2 poll would be performed less often, and a class 3 poll would be performed even less often. For each class data response, only the class data that has changed will be returned – keeping the response messages small and efficient. Finally, to acquire data not associated with either class 1, 2, or 3, an integrity poll, consisting of a class 0 scan, would be performed. Because of the possibly large amount of data that will be returned in a class 0 scan, it may not be terribly efficient and should be performed as least often as possible.

Message receive acknowledgement is supported in DNP3. However, this capability can be disabled to reduce networking overhead, albeit this must be used with caution because it can interfere with the normal behaviour of the system. For instance, unsolicited responses are such an example – because of their asynchronous, one-way nature, the absence of confirmation limits the capability of a Slave to detect a lost message.

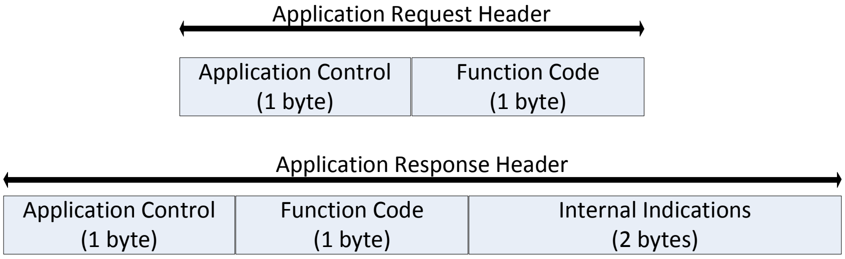
DNP3 is structured accordingly with the IEC EPA (Enhanced Performance Architecture) architecture, constituted by 3 layers:

- Application layer;
- Link layer;
- Physical layer.

DNP3 also adds an extra layer: the pseudo-transport layer, between the application and link layers, to deal with message fragmentation.

The **Application Layer** deals with the largest fragments the DNP3 stack can handle (2048 bytes maximum fragment buffer). If there is a need to transport more data within a single message, this layer is able to fragment it. APDUs (Application Protocol Data Unit) are created by joining the APCI (Application Protocol Control Information) and ASDU (Application Service Data Unit).

The Application Protocol Control Information (APCI) (Figure 2-8) is the message header while the ASDU is the payload. The APCI is different for request and response messages – the first ones are sent by Masters and include the Application Control (for flow control and fragmentation) and Function Codes to execute, while the second ones also include an Internal Indications field, used to report Slave errors and other state information.



Application Request Header

Application Control (1 byte)	Function Code (1 byte)

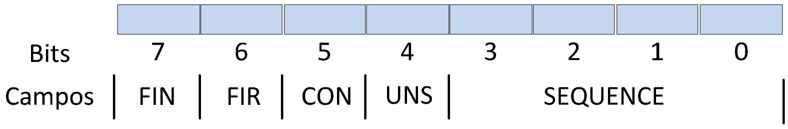
Application Response Header

Application Control (1 byte)	Function Code (1 byte)	Internal Indications (2 bytes)

Figure 2-8: DNP3 APCI Header [IEEE1815-2010]

The Application Control field (1 byte) is structured as such (see Figure 2-9):

- FIR – Active for the first message fragment;
- FIN – Active for the last message fragment;
- CON – When enabled, message confirmation is required;
- UNS – Enabled for Unsolicited Responses;
- SEQ – Sequence number, used for fragment reassembly;



Bits	7	6	5	4	3	2	1	0
Campos	FIN	FIR	CON	UNS	SEQUENCE			

Figure 2-9: Application Control field [IEEE1815-2010]

The **Pseudo-transport Layer** is located between the application and link layers. Its main purpose is to deal with fragmentation and reassembly in order to make messages fit in Maximum Transfer Unit (MTU) size of the physical technologies. In this layer, data payloads may have a size between 1 and 249 bytes, to which a 1 byte header is added (see Figure 2-10).

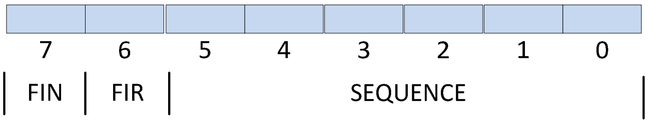


Figure 2-10: Pseudo-transport header structure [IEEE1815-2010]

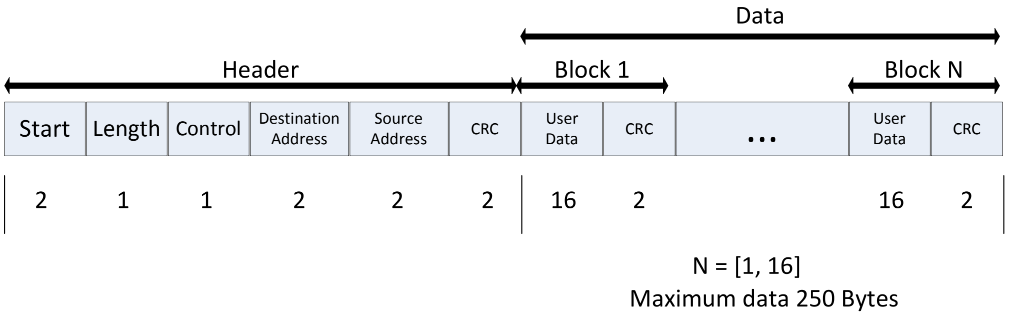
This header has the following structure:

- FIN – Enabled for the first fragment of a message.;
- FIR – Enabled for the last fragment of a message;
- Sequence – Fragment sequence number;

The **Link Layer** has the responsibility of managing everything related with the logical link between communicating parties, also dealing with device addressing. This layer adds addressing information and encapsulates frames for transmission – this is performed by adding a 10 byte header and a CRC-16 field per each 16 bytes of transmitted data. Figure 2-11 illustrates its structure:

- Start (2 bytes) - filled with 0x0564. Defines the start of the frame;
- Length (1 byte) – Length of the remaining elements of the segment, excluding error control bytes;
- Control (1 byte)- control byte, with the following structure:
    - DIR (Direction Bit) – Enabled if the segment was originated on a Master Station;
    - PRM (Primary Bit) – defines a primary (initial) or secondary (response) frame, also being used for interpretation of the FC of the APCI.
    - Frame Count Bits (2 bits) – two bits used on primary messages for detecting lost or repeated segments. When FCV (Frame Count Valid Bit) is active, the FCB (Frame Count Bit) bit toggles for each successful SEND-CONFIRM service that is initiated by the same primary station and directed at the same secondary station.
    - RES (Reserved) – Reserved, always 0;
    - DFC (Data Flow Control Bit) – this bit is used on secondary messages, to control the message flow in order to avoid buffer overflow issues. The Master will stop sending data and will send link state requests until receiving a message with the DFC bit zeroed;
    - FC (Function Code) – this field includes several link-level functions, whose significance depends of the segment being primary or secondary. It enables message reception confirmation, restarting connections, query the connection status, among others [Clarke2004].
- Destination Address (2 bytes);
- Source Address (2 bytes);
- CRC-16 control (2 bytes) – used for integrity verification;

- User Data – each block has 16 bytes size, with the last block size varying between 1 and 16 bytes. The size of the last block is limited to 10 bytes if the segment has the maximum allowed size;



Start	Length	Control	Destination Address	Source Address	CRC	User Data	CRC	...	User Data	CRC
2	1	1	2	2	2	16	2		16	2

Figure 2-11: DNP3 Link layer (Adapted from [IEEE1815-2010]

Finally, the physical layer defines the characteristics of the physical interface. It must provide the following functionalities:

- Connection;
- Disconnection;
- Sending;
- Reception;
- State reporting;

DNP/TCP

With the interconnection of SCADA networks to IP networks, a new version of DNP3 was developed especially for TCP/IP (Figure 2-12) with all the previously described DNP3 functions being put above the transport layer of the Open Systems Interconnection (OSI) model. Therefore, when using DNP3 for TCP/IP environments, fragmentation and reassembly functions performed at the OSI Application layer by the DNP3 Pseudo-Transport layer.

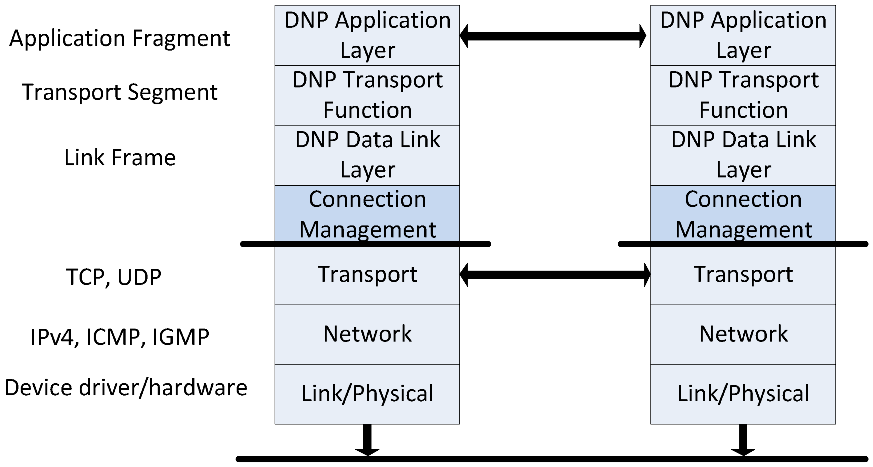


Figure 2-12: The DNP/TCP stack (Adapted from [IEEE1815-2010])

A new intermediate layer for Connection Management is added [IEEE1815-2010], to deal with the interface between the DNP3 layers and the TCP/IP stack. This layer establishes TCP connections, deals with sending and reception of UDP packets to and from the DNP3 upper layers.

IEC 60870-5-101/104

The IEC 60870-5 (IEC 60870 part 5) standard defines five transmission protocol documents for sending basic telecontrol messages between two systems, using permanent directly connected data circuits, and standard profiles necessary for uniforming applications, such as the IEC 60870-5-101, in which it is defined the way a device acts.

These documents are next briefly presented:

- IEC 60870-5-1 [IEC60870-5-1]: Specification of standards for coding, formatting and synchronizing data frames to be transmitted, of fixed and variable length, which meets specified data integrity requirements. These services are provided by the data link and physical layers for telecontrol applications.
- IEC 60870-5-2 [IEC60870-5-2]: Services for data link transmission using a control field and an optional address field (some point-to-point topologies do not require either the source or the destination address).
- IEC 60870-5-3 [IEC60870-5-3]: Rules for general structuring of application data in transmission frames, without specifying details about information fields and their contents. These rules are intended to be also used by a great variety of telecontrol application in the future.
- IEC 60870-5-4 [IEC60870-5-4]: Rules for the definition and coding of information elements, particularly digital and analogue process variables used in telecontrol applications.
- IEC 60870-5-5 [IEC60870-5-5]: Definition of basic application functions that perform standard procedures for telecontrol systems, situated between the Open System

Interconnection (OSI) application layer and the application program (see Table 2-2, grey shaded section). These functions are used for specific telecontrol task, as described later in detail, which result in the following application profiles, generated by the IEC Technical Committee 57 (Working Group 03):

- IEC 60870-5-101 [IEC60870-5-101]: Transmission protocols (basic telecontrol tasks).
- IEC 60870-5-102 [IEC60870-5-102]: Transmission of integrated totals in electric power systems (not widely used).
- IEC 60870-5-103 [IEC60870-5-103]: Transmission protocols (informative interface of protection equipment).
- IEC 60870-5-104 [IEC60870-5-104]: Transmission protocols (network access for IEC101).

Any functions that are not defined in the documents listed above must be specified within the profile. Examples of these functions are: station initialization, cyclic data transmission, data acquisition by polling and station configuration.

Unlike Modbus protocol, the IEC 870-5 standard (IEC 60870-5), as well as the DNP, is based on a three-layer reference model, used for efficient implementation within RTU devices, also defining basic application functionality for a user layer which adds interoperability for functions like clock synchronization and file transfer. This model is the Enhanced Performance Architecture (EPA), represented in Figure 2-13:

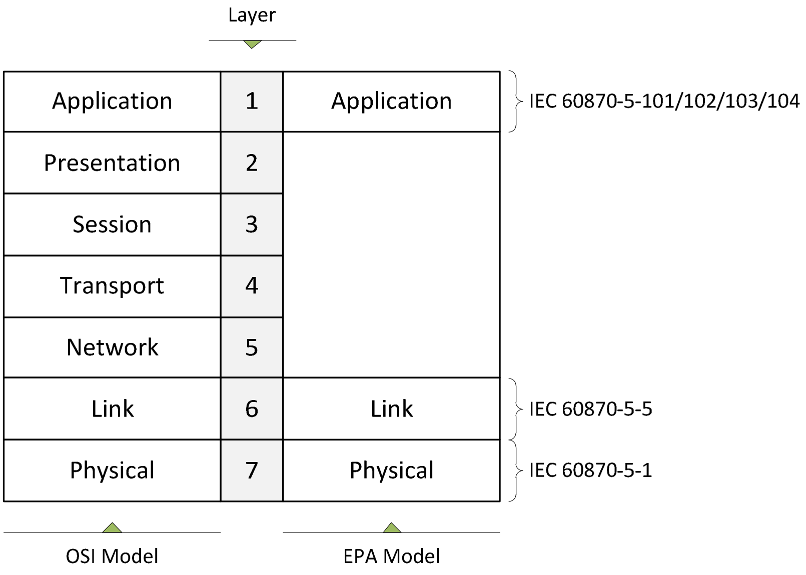


Figure 2-13: Reference models [TiangleMicroworks1999]

The main reason the EPA model only has 3 layers is to reduce the overhead of the 7-layer model (OSI), so it can be optimized for SCADA environments. As shown in the previous illustration, a correspondence is made between the OSI and EPA models to clarify where the layers match. Also, next to the EPA model references, at each of its 3 layers, is represented the location of the base documents and profiles, referred in the previous chapter.

As stated previously, the IEC104 standard is in fact an extension of the IEC101 to support TCP/IP connection, transporting IEC101 ASDUs (Application Service Data Units), based on the ISO-OSI reference model but only using 5 of these layers. So, the IEC101 standard is intended to work on serial RS232 lines, while the IEC104 standard, which is an extension of the previous, is intended to communicate over TCP/IP, with changes implemented on the transport, network, link and physical layers to enable such communications. The application layer is maintained in both standards. The IEC 60870-5-104 reference model is located in the application layer, as mentioned in Table 2-2 below, and in Figure 2-14, later, where the APDU (Application Protocol Data Unit) definition is illustrated.

Table 2-2: IEC 60870-5-101/104 network reference model [Weiqing2010a]

Layer	Description
User layer[2]	Selected application functions of IEC 60870-5-5: a) Station initialization b) Cyclic data transmission c) General interrogation d) Command transmission e) Parameter loading f) File transfer g) Data acquisition by polling h) Acquisition of events i) Clock synchronization j) Transmission of integrated totals k) Test procedure
Application layer (7)	Selection of ASDU from IEC 60870-5-101 and 104
	> Application Protocol Control Information (APCI) > Transport Interface (User to TCP interface)
Transport Layer (4)	Selection of TCP/IP Protocol Suite (Request for Comments (RFC) 2200)
Network Layer (3)	

[2] User layer does not correspond to a real layer of the OSI model. It is just a representation to better understand the application functionality defined in IEC 60870-5.

Link Layer (2)	
Physical Layer (1)	

As seen in the previous Table 2-2, the IEC104 protocol is a combination of IEC101 and the network transmission function provided by TCP/IP, to allow the later to be used with some of the existing TCP/IP network types. Also, in a green scale are the introduced layers in IEC104 that are not present in IEC101, since only the first adds support for TCP/IP connection.

In respect to the 3 layers referent to the IEC101 protocol, following is a description of what each one of these represent:

- *Application Layer* - Selected application information elements of IEC 60870-5-4 for definition and coding of information elements and the Application Service Data Units (ASDUs) of IEC 60870-5-3 for general structure of application data. The contents and sizes of individual information fields of the ASDUs (see Figure 2-19) are specified according to the declaration rules for information elements defined in IEC 60870-5-4. Also, type information defines the structure, type and format for information objects.
    - These 2 predefined parameters (elements and type information) do not allow the addition of new information elements or types by any vendor. In fact, the information elements have been defined for equipment protection, voltage regulators and for meter values to interface Intelligent Electronic Devices (IEDs) with the RTUs.
- *Link Layer* - Selected link transmission procedures of IEC 60870-5-2 for data link transmission services and the transmission frame formats of IEC 60870-5-1. The transmission mode (balanced or unbalanced) is also defined in this layer as well as the provided addresses for each link.
- *Physical Layer* - Selected International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) recommendations, defining the hardware-dependent specifications of the IEC 60870-5-101 and 104 communication interfaces, compatible with Electronic Industries Association (EIA) standards RS-232[3] and RS-485[4], also supporting fiber optic interfaces.
    - The IEC 60870-5-1 standard offers the asynchronous FT 1.2 frame format, specified in IEC101, to provide data integrity with the maximum efficiency for acceptable convenience of implementation, using standard Universal Asynchronous Receiver/Transmitters (UARTs).

IEC104 protocol provides 255 bytes APDU packets (including start character and length identification), meaning that the ASDU maximum length is 253. Also, the APDU length

[3] Interface between data terminal equipment.

[4] Electrical characteristics of generators and receivers used in balanced digital multipoint systems.

includes 4 octets of control field and ASDU, meaning that the maximum ASDU length is 249. So, this type of provision limits an APDU packet to send up to 121 normalized measured values without the quality descriptor or a 243 single-point information data, otherwise, if the amount of collected data by an RTU exceeds the above limit, the APDU packet has to be divided before being sent. The APDU packet structure is illustrated in Figure 2-14, as previously mentioned.

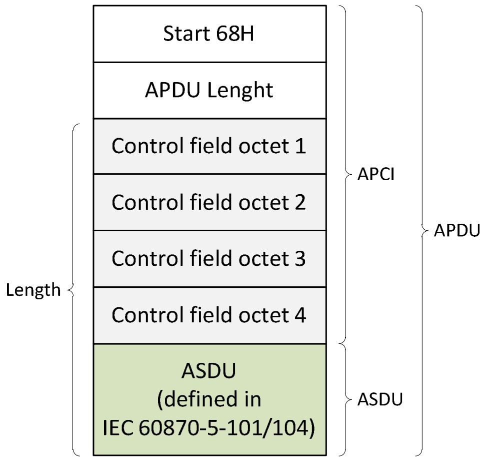
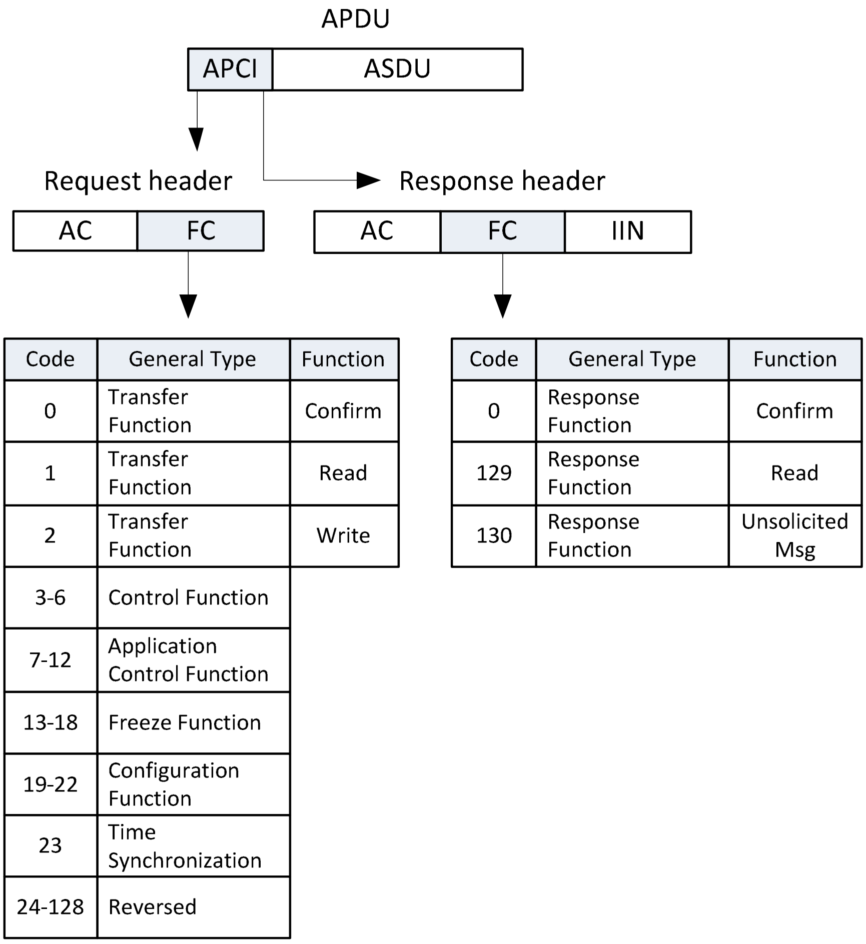


Figure 2-14: APDU Packet [Weiqing2010b]

The application header, also illustrated in Table 2-2 (at the Application layer), is referred to as the Application Protocol Control Information (APCI), which may be either 2 or 4 bytes, depending whether it is a request or a response, as shown in Figure 2-15, where it is also illustrated the Function codes inside each one of these.



Code	General Type	Function
0	Transfer Function	Confirm
1	Transfer Function	Read
2	Transfer Function	Write
3-6	Control Function	
7-12	Application Control Function	
13-18	Freeze Function	
19-22	Configuration Function	
23	Time Synchronization	
24-128	Reversed	

Code	General Type	Function
0	Response Function	Confirm
129	Response Function	Read
130	Response Function	Unsolicited Msg

Figure 2-15: Application function codes [Gordon2004]

The keyword AC stands for Application Control, which has a corresponding Function Code (FC) and type.

The control fields of the APDU in Figure 2-14 define the control information for protection against message loss or duplication, start and stop of message transfers and for supervision of transport connections. These octets can be classified into three kinds of message formats by its definition.

- **I format** (Numbered information transfer)

This filed is used for APDUs containing an ASDU, i.e., information being indicated by a 0 (zero) in the first bit position. The frame is then represented in Figure 2-16.

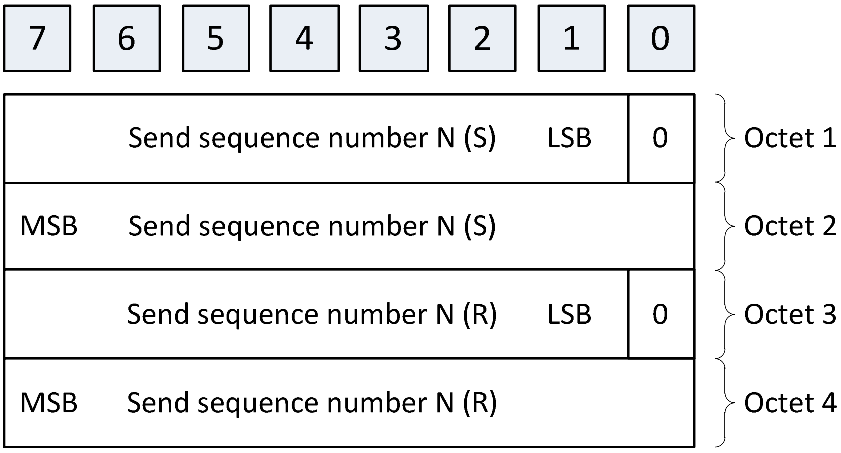

7
6
5
4
3
2
1
0
Send sequence number N (S)
LSB
0
Octet 1
MSB
Send sequence number N (S)
Octet 2
Send sequence number N (R)
LSB
0
Octet 3
MSB
Send sequence number N (R)
Octet 4

Figure 2-16: Information (I) format control field (Variable length frame) [Lian2011a]

- **S format** (Numbered supervisory functions)

This filed is used for APDUs containing only the APCI header. Unlike the previous, these frames do not have any information attached and so, are only used for controlling the transport of the APDUs. It is indicated by 1 in the first bit position followed by a 0 (zero) in the second bit position. The frame is represented in Figure 2-17.

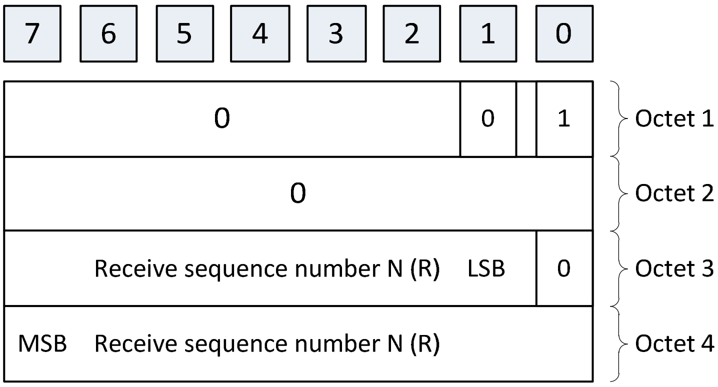


Figure 2-17: Supervisory (S) format control field [Lian2011b]

- **U format** (Unnumbered control functions)

Just like the previous, this field is also used in APDUs that only contain the APCI. It is used as a start-stop mechanism for information flow or when more than one connection is available between stations, also allowing a changeover between these connections without losing data. Note that there are no sequence numbers. The frame is represented in Figure 2-18.

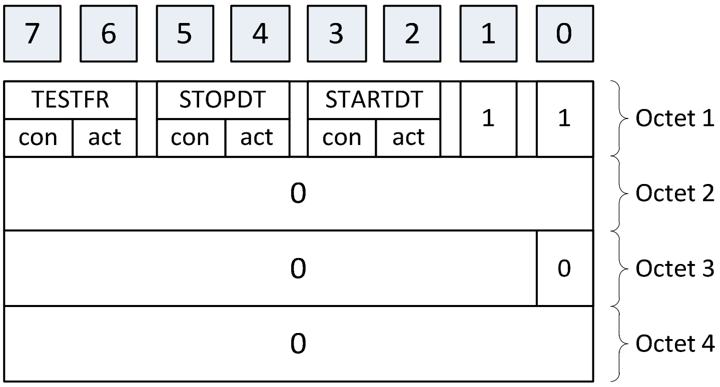


Figure 2-18: Unnumbered control (U) format control field [Lian2011c]

The sequence numbers used in these control fields are used to control the APSUs flows in both directions. Once the receiver gets an APDU, it advertises to the sender of the highest sequence number, using an I or S format message, so the sender can re-send ASDUs that may have got lost. This also depends on whether receiver is sending information in the opposite direction.

A detailed view of the ASDU frame (green shaded section in Figure 2-14) is represented in Figure 2-19 with the respective fixed and variable fields, as in the IEC 60870-5-1 document and at the “Application layer” section above.

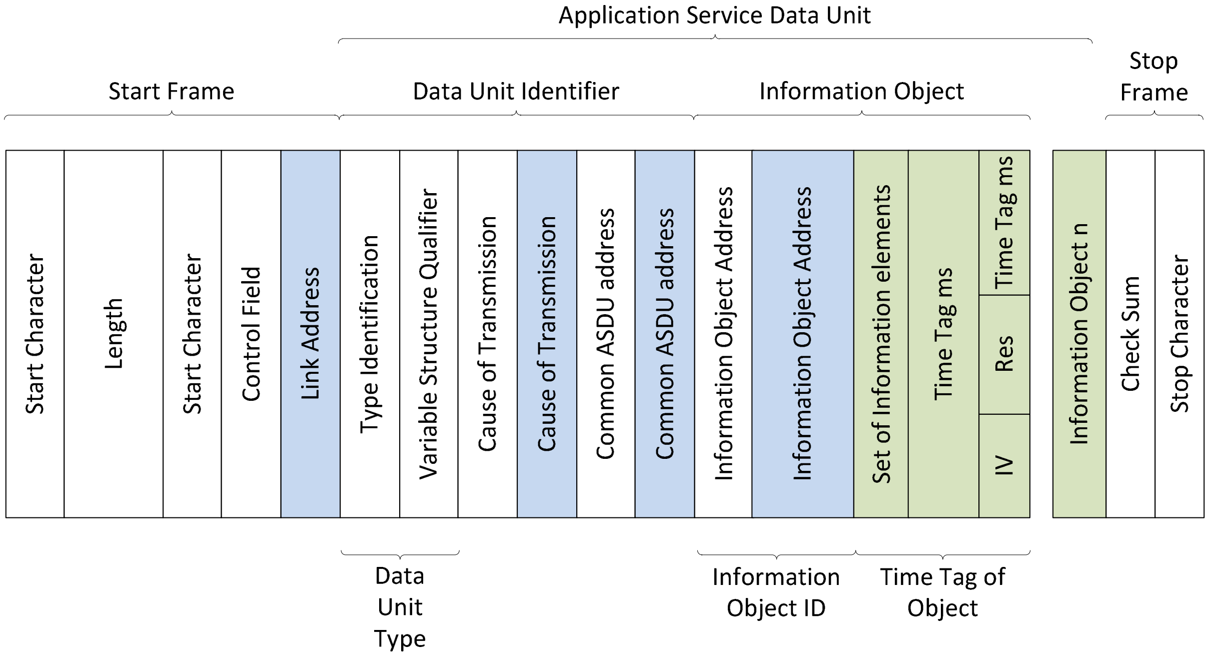


Figure 2-19: ASDU Frame [Jay2003a]

In Figure 2-19, blue shaded sections are the optional fields which will be determined by a system level parameter shared by all devices in the system. The green shaded sections are the variable fields per ASDU, whose common address size is determined by a fixed system parameter, in this case 1 or 2 octets (bytes). The remaining fields are all fixed per ASDU.

The standard 101 (IEC10) profile has 2 definitions that are not present in any of the documents previously referred:

- *Control direction* – Transmission from the controlling station to the controlled station.
- *Monitor direction* – Direction of the transmission from the controlling station to the controlled station.

In order to aid administrators in the configuration of their SCADA systems, the 101 profile defines a check list in which these can ensure interoperability between the used devices and the ones from other vendors. This list does not only contains information from the ASDU for both control and monitor direction (as previously referred), but also parameters such as baud rate, ASDU field length common address, link transmission procedure, basic application functions defined in IEC 60870-5-102 and 105 documents. This check list allows vendors to define their devices or system in a protocol perspective.

When communicating, both devices in the SCADA system using the IEC 60870-5-101 protocol can perform its transmissions in 2 different modes: balanced and unbalanced. At the data link layer, the standard 101 profile species whether an unbalanced (includes multi-drop) or balanced (includes point-to-point) transmission mode is used together with which link procedures (and corresponding link function codes) are to be used. Also specified is an unambiguous number (address) for each link.

- **Unbalanced mode**

In this case, only the Master station can initiate a transmission, polling the controlled outstations, which can only respond when the requests are sent by the first. The supported transmission service types selected from IEC 60870-5-2 are described in Table 2-3, as follows:

Table 2-3: Service types initiated by the Master station [ABB2010a]

Service	Description
SEND / NO REPLY	Global messages and cyclic set-point commands for the Master station
SEND / CONFIRM	Control and set-point commands from the Master station
REQUEST / RESPOND	Data polling from the controlled outstations

- **Balanced mode**

Unlike the previous case, here any station involved in the communication can initiate the transmission of messages, acting as controlling (Master) stations or controlled outstations, simultaneously. Therefore these devices are called combined stations, being restricted to point-to-point and to multiple point-to-point configurations. Table 2-4 describes the supported transmission services are described.

Table 2-4: Service types initiated by controlling and controlled stations [ABB2010b]

Service	Description
SEND / NO REPLY	Global messages and cyclic set-point commands
SEND / CONFIRM	Control and set-point commands

The first type of service (SEND/NO REPLY) mentioned in Table 2-4 above can only be initiated by a controlling station with a broadcast address in a multiple point-to-point configuration.

Following is a list of basic application functions implemented by the current standard, as it has been referred:

- *Data acquisition* – Since the data may appear faster than the communication link is able to transfer, the controlled station buffers all data, such as, command replies or process values collected cyclically, upon change or request from the Master station. The actions performed on the buffered data varies whether balanced or unbalanced transmission is used: For unbalanced transmission, on the link layer the controlled stations wait for a request coming from the Master station, which polls the buffered data. On the other hand, for balanced transmission, the controlled station transmits the data to the Master station without a delay.
- *Event acquisition* – The events occur at the controlled station's application level, being also buffered for the same reasons mentioned for Data acquisition.
- *Interrogation* – This function is used to update the controlling station after an internal station initialization or when an information loss is detected, being performed either by an interrogation group (1-16) at a time or all groups at once. When requested, the controlled stations transmit the actual values of their process variables.
- *Clock synchronization* – After the clock of the controlled station is synchronized with the one on the controlling station, it keeps synchronizing periodically with the C_CS ACT command. This provides a correct chronological sequence of time-tagged events or information objects. When an ASDU is received, the time information must be corrected by one of the end devices. Also, the transmission delays are calculated by a delay acquisition command so the time is corrected at the controlled station when sending.
- *Command transmission* – In order to change the state of the operation equipment, a

command may be sent by the controlling station which can be one of the following:

  - *Direct command* – Used to immediately control operations in controlled stations. For safety purposes, the permissibility and validity of the received messages are checked.
  - *Select and execute command* – Used to prepare a specified control operation in a controlled station, check if the correct control operation is prepared and, finally, execute the command. In this case, the preparation is checked by an operator or by an application procedure and if the controlled station does not receive the correct execute indication, the control operation does not start. The controlled station receives a command transmission confirmation through an activation confirmation response and after the command is executed, an activation termination response is sent to the controlling station.

- *Integrated totals transmission* – An integrated total is a value that is integrated over a specified period of time. In the other hand, a system parameter corresponds to the specific clock times and the periodic time interval of successive acquisitions of the integrated totals. Two methods for acquiring counter information are: Acquisition of integrated totals (Freeze-and-read); and acquisition of incremental information (Clear-and-Read).
- *Protocol and Link parameters changes* – When changed, the new values of the protocol and link parameters take effect after they have been committed.
- *Transmission delay acquisition* – Time correction is determined by the sum between the transmission delay and the internal equipment delay. To obtain the value of the transmission delay, either parameterization or using a dynamic procedure (initiated by the controlling station) are both valid alternatives.
- *Analog Value Deadband* – The use of the deadband feature allows a user to reduce the number of unnecessary events using analogue measurements for each point, which might be configured using proper tools, by setting 2 parameters:
  - *Range* – Considering a range of 0.05 (5%), if the data point value changes beyond this value from the previously sent one, the data will be sent as deadband data.
  - *Interval* – Limits the deadband value to be sent once per configured time window, in seconds. To disable this feature, the interval is set to 0.

2.4.1.6 Comparison of IEC101, DNP 3 and Modbus

With the Modus, DNP 3 and IEC104 protocols already described, it is interesting to check a comparative table in which their features are briefly described. Table 2-5 describes a comparison based on the features of each protocol. Some of the most relevant information in this table includes, features by layer (Physical, Data link and Application layers), addressing, required parameters, application specific information, etc.

Table 2-5: SCADA Protocols comparison [Jay2003b]

Feature	IEC 870-5-101	DNP 3.0	Modbus
Standardization	IEC Standard (1995) Amendments 2000,2001	Open industry specification (1993)	Not Applicable
Standardization Organization	IEC TC 57 WG 03	DNP users group	Modicon Inc.
Architecture	3-layer EPA architecture	4-layer architecture Also supports 7 layer TCP/IP or UDP/IP	Application layer messaging protocol
Physical layer	Balanced Mode - Point to Point Multipoint to point Implementation by X.24 / X.27 standard Unbalanced Mode - Point to Point to Multipoint Implementation by V.24 / V.28 standard	Balanced mode transmission It supports multiple masters, multiple slave and peer-to-peer communication RS 232 or RS 485 implementation TCP/IP over Ethernet, 802.3 or X.21	Balanced mode of transmission RS 232 serial interface implementation Peer to peer communication TCP/IP over Ethernet
Data link layer	Frame format FT 1.2 Hamming distance - 4	Frame format FT3 Hamming distance-6	Two types of message frames are used: ASCII mode and RTU mode
Application layer	Both IEC 870-5-101 and DNP 3.0 provides: > Time synchronization > Time stamped events > Select before operate > Polled report by exception > Unsolicited responses	Remote starting / stopping of software applications Polling by data priority level Broadcast addressing	Does not give time stamped events. We have sequence of events (without time but not event list with time. Does not provide polled report by exception

	> Data group/classes Limited to single data type per message Can control one point per message only No internal indication bits No application layer confirms for events	Multiple data types per message are allowed Internal Indication field IID present in response header Application layer confirms events; use of CON bit is made	Checksum ensures proper end-to-end communication
Device Addressing	Link address could be 0, 1, 2 bytes Unbalanced link contains slave address Balanced link is point to point so link address is optional (may be included for security)	Link contains both source and destination address (both always 16 bits) Application layer does not contains address 32 b point addresses of each data type per device	Addresses field contains two characters (ASCII mode) or 8 bits (RTU mode) Valid address in range 1-247 Address 0 used for broadcast
Configuration Parameters required	Baud rate Device addresses Balanced / unbalanced Frame length Size of link address Size of ASDU address Size/structure of point number	Baud rate Device addresses Fragment size	Baud rate Mode ASCII or RTU Parity mode

	Size of cause of transmission		
Application Specific information model	A few application specific data types available Data objects and messages are not independent to each other	Permits vendors to create application specific extensions Data objects and messages independent to each other	Allows user to create application specific model
Cyclic transmission	Eliminates static data poll message from master Interrupted by event triggered communication request	Available but interval cannot be remotely adjusted	Not Applicable
Dominant market	Europe (South America, Australia and china)	North America (Australia and china)	Used worldwide for application with low volume data
Online configurations	Enable/ disable communication control objects Loading configuration Change report / logging behaviour	Define group of data Selecting data for responding Enable/ disable communication control objects Loading configuration Change report / logging behaviour	Efficient online configuration could be made by Modbus TCP/IP
Open for other encoding solutions	Not Available	Yes open for other encoding solutions like XML	Yes. One could write source code in programming languages such as C, VC++ & JAVA etc.

The protocol selection depends on the scenario and the operator expectancies. There isn't a best protocol for every situation. With this in mind, some issues must be considered in order to choose a proper solution, as those presented below:

- *Application domain* – When dealing with utilities or oil and gas industries, operators should go either with IEC101/104 or DNP 3, especially if the SCADA system has requirements such as time-stamping. Modbus is more of a general-purpose solution, being more suited for industrial applications with direct register mapping and small volumes of data.
- *Communication devices* – Depending on which are the communicating devices, one might have one of the following situations: If communication with substations there are protocols meant for protection control and metering, such as Modbus, IEC103 or Profibus [Profibus]; if the communication is established outside substations, protocols used for the exchange of data between substations and control centers are IEC101/104 or DNP 3; For communications between applications, there is the IEC 61968 standard, still under development.
- *Specific requirements* (e.g. amount of data, bandwidth, response time and distance between devices) – When sending large volumes of data, both DNP 3 and IEC101 present good solutions. However, if there is need to transmit huge volumes of data across long distances in serial links working with high baud rates, DNP 3 is favoured protocol over IEC101. If a simple setup is to be used, Modbus is probably more adequate since it requires less memory, has fewer data types, smaller frame sizes, imposing less overhead.
- *Devices to equip* (e.g. Embedded devices, PLCs, Personal Computers (PC)) – smaller overhead makes Modbus more adequate for embedded controllers with limited capabilities.
- *Interface functions* (e.g. Parameterize relays remotely, download disturbance data and events, retrieve measurements) – for simple and low-complexity data interfaces, Modbus might be adequate, although for more complex scenarios other protocols might be considered.
- *Domain players* – The manufacturer and model of the device also limits choice to the list of the supported protocols.
- *Geographical location* – For example, in power systems, some protocols are more popular in specific geographic areas than others – for instance, for deployments in Europe, the most obvious choice would be IEC101/104, while in North America, DNP3 would be used instead.

Complementary to this chapter, Chapter 7 provides an appendix with a brief synthesis of SCADA protocols in the scope of the energy production and distribution industry.

3 Vulnerabilities of SCADA Systems

3.1 Historical evolution and impact on security of SCADA systems

Initially, ICS systems were isolated by nature, being limited to the process network – in those times, security was guaranteed by both obscurity and isolation. Protocols were proprietary and its documentation was undisclosed, creating a false sense of security [Clarke2004]. Only manufacturers and attackers knew of failures and vulnerabilities, with both parts having no interest in their divulgation.

Still, “modern” SCADA architectures are, in general, much similar to the ones used in the ‘80s and ‘90s, even if some technologies suffered a clear evolution. This has to to with several reasons such as maturity (these architectures are tried and tested) and cost of migrating to a modern solution

Unfortunately, when migrating from an “isolated” to an open environment, from serial communication to TCP/IP communication, conventional SCADA architectures started to show all their limits. The move to more open scenarios with network interconnection together with the use of ICT technologies and the increasing adoption of open, documented protocols, exposed serious weaknesses in SCADA architectures.

By itself, the growing trend towards the interconnection of the ICS network with the ICT infrastructure, and even with the exterior (for instance, for connection with internal company systems or for remote management) created a new wave of security problems and attacks. In fact, there is a growing trend in the number of externally initiated attacks on ICS systems, when compared with internal attacks [Kang2011].

Also, the adoption of commercial operating systems brought its own share of problems. Albeit reducing development and lifecycle management costs, the adoption of these operating systems made SCADA infrastructures implicitly vulnerable to a vast array of issues that traditionally plague them. There are several security incidents and undirected attacks to SCADA infrastructures that were the result of operating system vulnerabilities.

The security by obscurity philosophy (which is not a good security practice, anyway) became unfeasible. However, the problem of security in SCADA systems was ignored for several years, and even now serious issues persist. For instance, unsafe protocols such as Modbus are being widely used in production systems. But even new features, such as the auto-configuration capabilities of certain equipment (plug-and-play) only got things worse, since attackers found it to be a valuable resource [Clarke2004] for attack planning and execution.

Also, the old-school mindset still persists, up to the point that some process managers still think of ICS systems as isolated and implicitly secure [Krutz2006], disregarding the need for regular security updates or software patching procedures, increasing the probability of a successful attack. Still, software updates might be difficult, for other reasons: the fact that some components have to work on a continuous basis without interruptions, up to the point of working years without being reinitialized [ESCoRTS2010c] [Zhu2011]; due to the fact that any software release must be carefully tested by equipment manufacturers before being released, or even due to end-of-life support for specific devices or software frameworks.

There are standards and good practice guidelines on the implementation and operation of SCADA systems, such as the American National Standards Institute ANSI/ISA-99.00.01-2007 [ISA-99.00.01] or AGA-12 (from the American Gas Association) [AGA12]. Nevertheless, such guidelines are rarely adopted or closely followed in real production environments. Aditionally, the limitiations of already existing systems reduce the efectiveness of such orientations, since they constitue a source of unresolved several security issues which are difficult deal with.

3.2 ICT vs. ICS systems

Initially [NIST], ICS had little resemblance to ICT systems in that ICS were isolated systems, running proprietary control protocols using specialized hardware and software. Widely available, low-cost Internet Protocol (IP) devices are now replacing proprietary solutions, which increases the possibility of cyber security vulnerabilities and incidents. As ICS are adopting ICT solutions to promote corporate connectivity and remote access capabilities, and are being designed and implemented using industry standard computers, operating systems (OS) and network protocols, they are starting to resemble ICT systems. This integration supports new ICT capabilities, but it provides significantly less isolation for ICS from the outside world than predecessor systems, creating a greater need to secure them.

Figure 3-1 shows an isolated ICS and Figure 3-2 shows an integrated ICS with a corporate network.

Particularly, in Figure 3-1, differently from Figure 3-2, there aren't links between the corporate LAN and the control system network. The isolated system is de facto immune to the attacks that come from the Internet; obviously is always possible to attack this system by introducing the malware with a portable device, but this kind of vector is always possible.

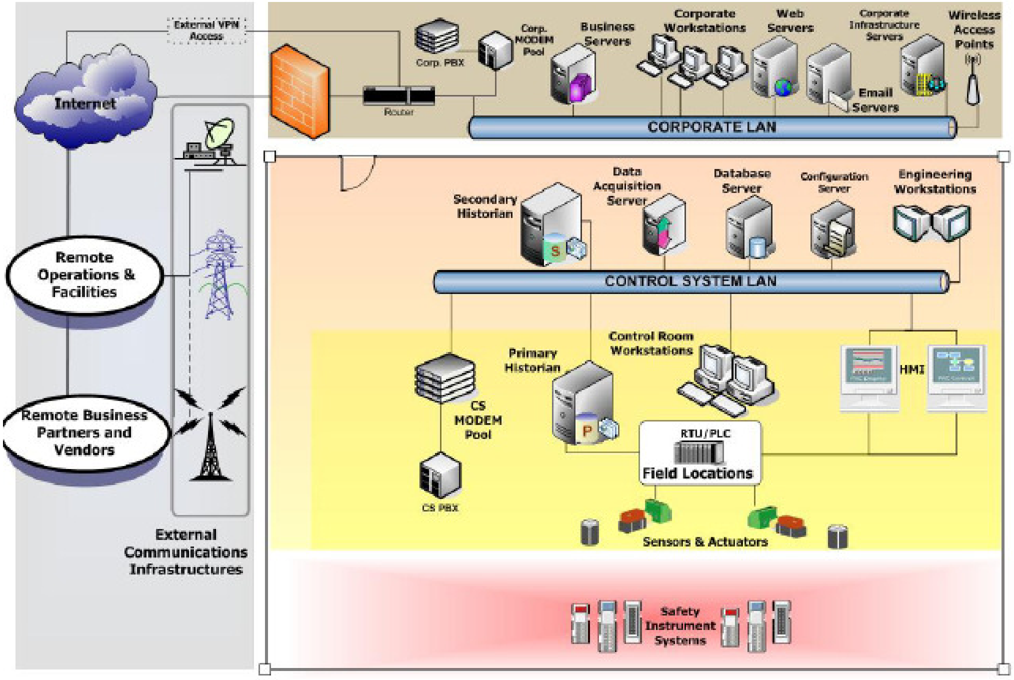


Figure 3-1: Isolated ICS [Homeland]

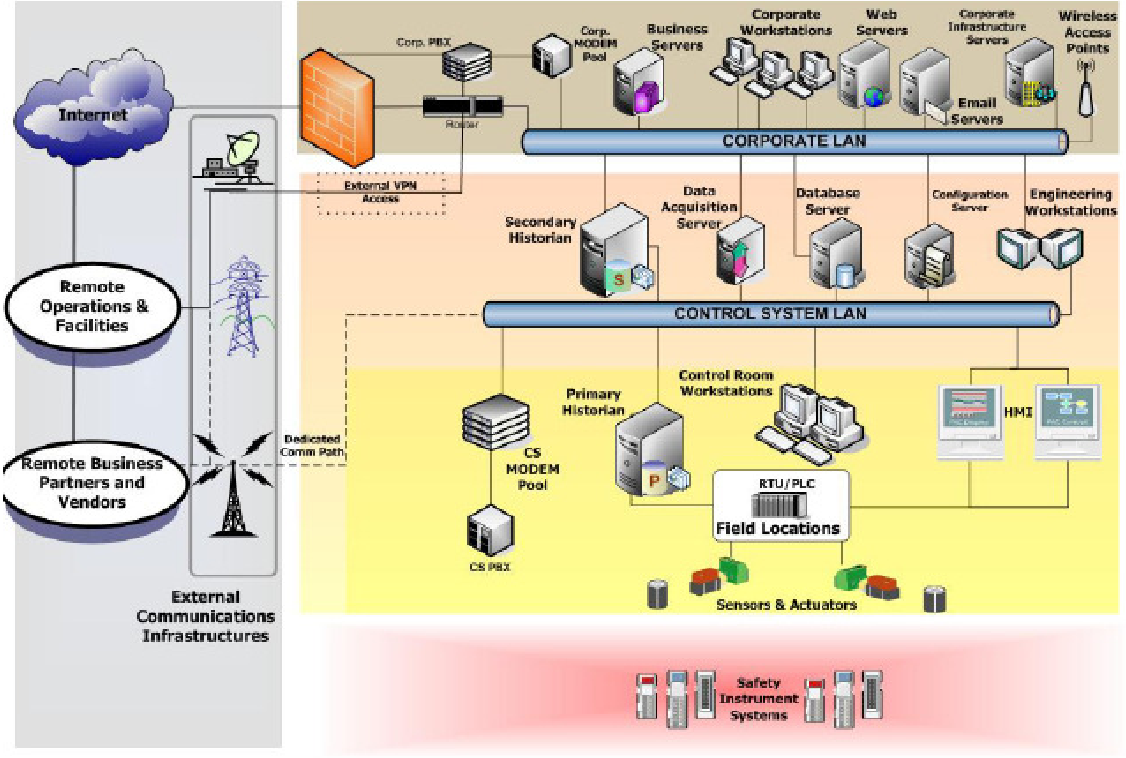


Figure 3-2: Integrated ICS and corporate network [Homeland]

ICS are intrinsically unsecure, as discussed in the previous section. There are two distinct threats to a modern ICS:

- The first one is the unauthorized access to control software, installed in any ICS device. It could be an unauthorized human access or unauthorized changes induced intentionally or accidentally by virus infections and/or other malicious software residing on any control device.
- The second one is the packet access to the network segments hosting ICS devices. In many cases, there is rudimentary or no security on the actual packet control protocol, so anyone who can send packets to any ICS device can control it. In many cases, ICS users assume that a Virtual Private Network (VPN) is a sufficient protection and are unaware that physical access to ICS-related network jacks and switches provides the ability to totally bypass all security on the control software and fully control those ICS networks. These kinds of physical access attacks bypass firewall and VPN security and are best addressed by endpoint-to-endpoint authentication and authorization such as is commonly provided in the non-ICS world by in-device Secure Sockets Layer (SSL) or other cryptographic techniques [GAO2005].

Many vendors of ICS and/or control products have begun to address the risks posed by unauthorized access by developing lines of specialized industrial firewall and VPN solutions for TCP/IP-based ICS networks as well as external ICS monitoring and recording equipment. Additionally, application whitelisting solutions[5] are being implemented because of their ability to prevent malware and unauthorized application changes without the performance impacts of the traditional antivirus scans.

The increased interest in ICS vulnerabilities has resulted in vulnerability researchers discovering vulnerabilities in commercial ICS software and more general offensive ICS techniques presented to the general security community. In electric and gas utility ICS systems, the vulnerability of the large installed base of wired and wireless serial communications links is addressed in some cases by applying bump-in-the-wire devices that employ authentication and Advanced Encryption Standard (AES) encryption rather than replacing all existing nodes.

By this method a hardware device that provides Internet Protocol Security (IPSec) services is added. For example, supposing a company with two sites, each one that has a network that connects to the Internet using a router that is not capable of IPSec functions, a special “IPSec” device between the router and the Internet at both sites is interposed, as shown in Figure 3-3. These devices will then intercept outgoing datagrams and add IPSec protection to them, and strip it off incoming datagrams.

[5] A whitelist is a table that contain all the device that are always considered to be safe

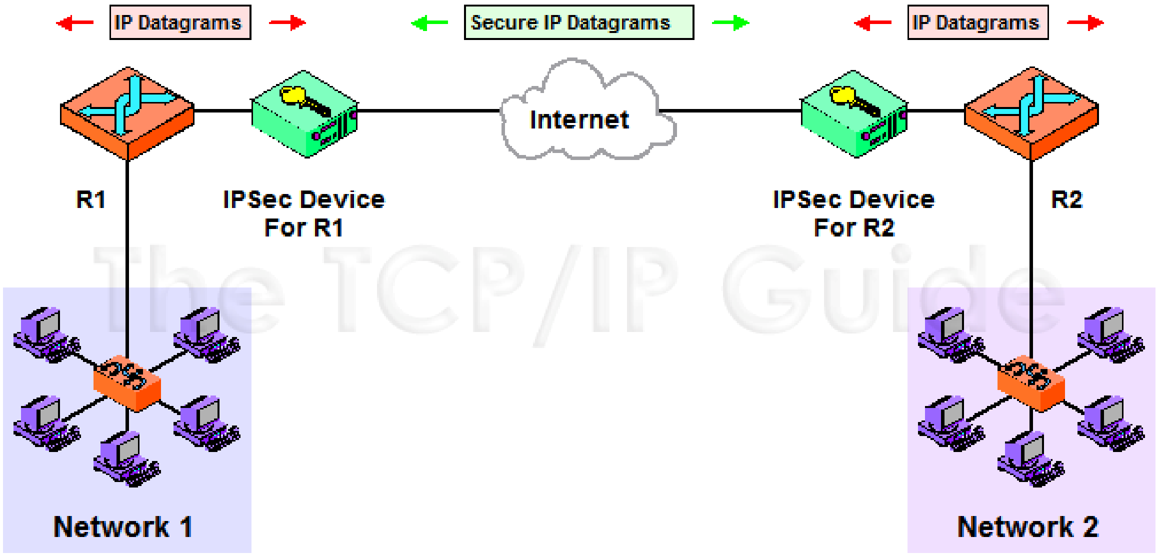


Figure 3-3: IPSec “Bump In The Wire” (BITW) Architecture

Critical Infrastructure cyber vulnerabilities involve the enterprise/corporate network, the industrial control systems and the critical infrastructure itself (i.e. power system). Figure 3-4, where Industrial Control systems are part of Process Control Network, and Figure 3-5, where ICS is specified as SCADA, show how they are linked together. Both Figure 3-4 and Figure 3-5 come from [Chiesa2009]. Figure 3-5 also adds a component view of corporate network and SCADA. In such a configuration it is very simple for an attacker to go from the enterprise/corporate network to the ICS process control network because he can reach every points of the SCADA network from every computer of the enterprise/corporate (ICT based) network and there is only a switch between them (a switch doesn’t provide any kind of authentication or security policy). Many of the protection measures used in standard ICT security frameworks (firewalls, IDSs and other) can be adapted in the process control & SCADA environments.

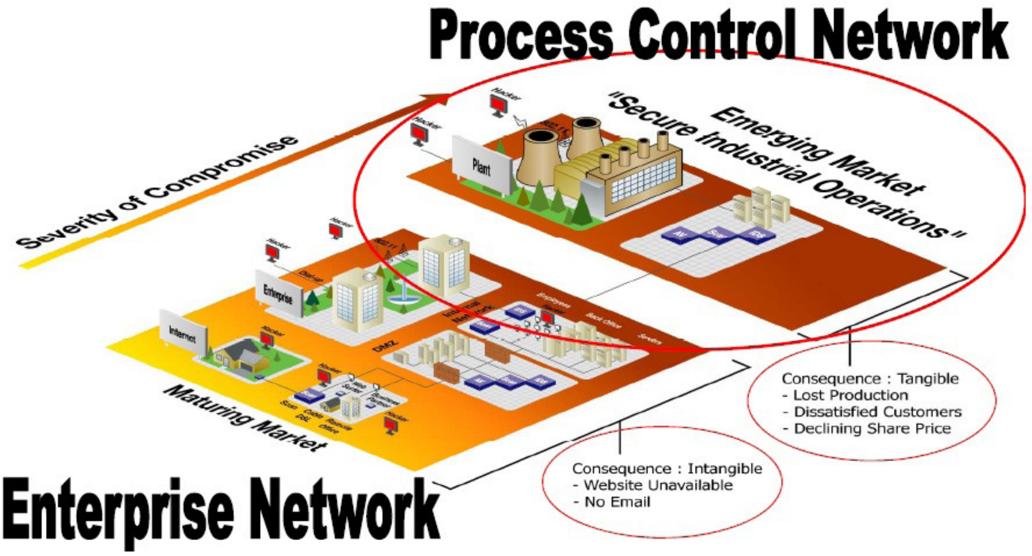


Figure 3-4: External view with the enterprise network linked with the process control network [Homeland]

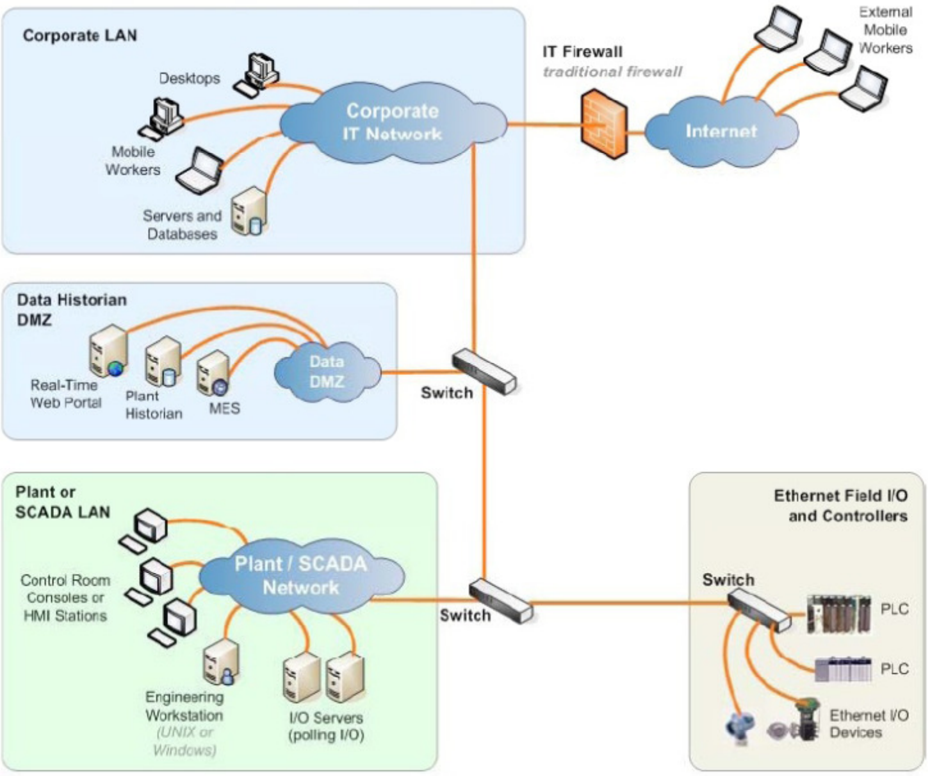


Figure 3-5: Component view of corporate network and SCADA [Homeland]

However, when adopting ICT mechanisms or technologies, for use in ICS environments, special care must be taken. While the adoption of COTS components (hardware and software) is seen as a way to reduce costs and development/deployment times, it also has the drawback of introducing some security risks, mainly because there are some assumptions regarding ICT networks that not always are equally true in ICS environments.

As an example, we have traditional firewall appliances, used in ICT environments. Some of these appliances assume that they are place behind a router capable for blocking TCP SYN packet floods. Therefore, several of these appliances do not implement any kind of protection against these attacks [Byres2005]. Figure 3-6 shows an example of this situation.

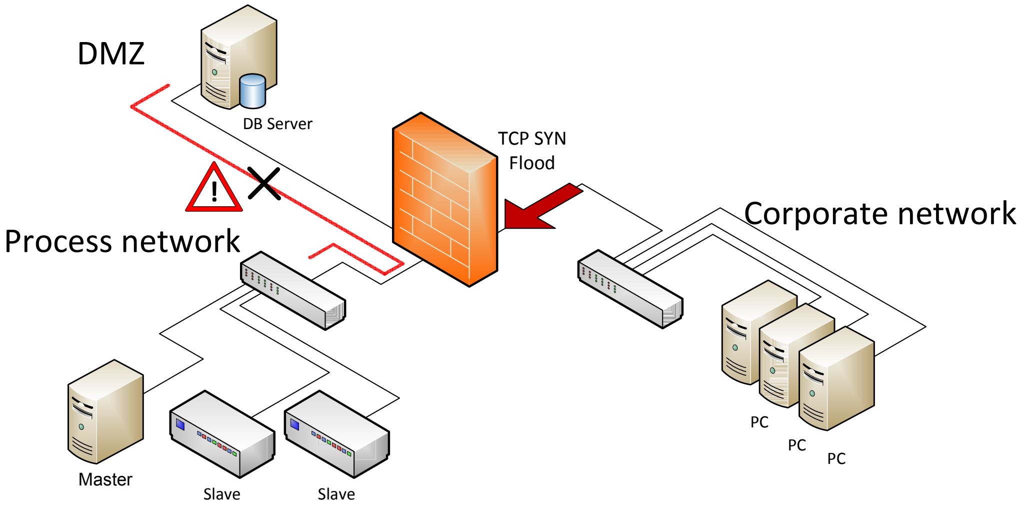


Figure 3-6: TCP SYN Flood scenario (Adapted from [Verba2008])

In a firewall with three interfaces, we have the corporate network, the process network and the DMZ (DeMilitarized Zone). If a TCP SYN Flood attack is started from a compromised host on the corporate network, it is possible that the process network to lose connection with the database server on the DMZ, used for supporting the SCADA platform.

Also, command execution latency is a very important matter for SCADA systems, since it has an impact on the real world. These systems frequently have demanding availability and response time requirements, frequently with little tolerance for latency or delay. In ICT environments, there are soft real-time applications, such as VoIP (Voice over IP), where latency requirements are low (ideally with communication latencies below 150ms), but where it is possible to discard packets or adjust encoding parameters to compensate for delays. SCADA applications, on the other hand, are generally classified as hard real-time, with strict latency and delay limits. An excessive delay in the execution of a critical command can potentially cause damage, equipment destruction or even human loss, in extreme cases [Zhu2011].

It is very important to keep in mind that ICS systems have a different set of priorities, when compared with ICT infrastructures – to a certain extent; this inversion is one of the causes of the security problem with SCADA infrastructures. Figure 3-7 illustrates this situation.

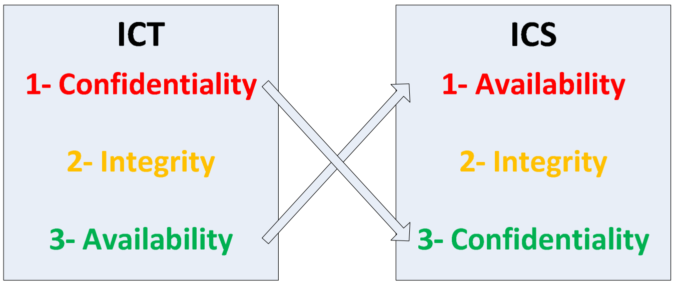


Figure 3-7: ICT vs. ICS priorities (Adapted from [ISA-99.00.01])

On ICT networks, confidentiality and security is maximum priority, followed by communications integrity and, finally, by availability. For (ICS)/SCADA, on the other hand, is a different matter because of their critical nature, which causes an inversion of priorities [ISA-99.00.01]. This priority difference has a real impact when it comes to choose and implement security mechanisms, even more if they are imported from the ICT world.

Product lifecycle is another matter which is different on both worlds – ICT infrastructures have substantially shorter lifecycles, when compared with their ICS counterparts. In ICT infrastructures, equipments and systems are renewed from time to time, something that contrasts with the ICS philosophy of using mature systems, sometimes far beyond their projected lifetime. This limits the possibility of implementing some security mechanisms due to the limited capabilities of existing equipment [Igure2006].

3.3 Infrastructure vulnerabilities and threats

3.3.1 Overview

The necessity of interconnections among Industrial Control Systems and the related enterprise systems (Energy Management Systems, Distribution Management Systems and Substation Automation, etc.) is being emphasized and, as a consequence, standard operating systems such as Windows and UNIX, public networks such as Internet and general communication technology such as wire and wireless networks are widely used.

Among the main ICS vulnerabilities in existence, in the following there is a list of representative vulnerabilities of a SCADA system [Chiesa2009, CRS2008, Ryu2007].

- Diversity of vendors : different characteristics of each vendor’s
- SCADA work process and various protocols and operating systems.
- Widening of networks: difficulty of network management due to the facilities being scattered over a large area.
- Aging of equipment: when most installed equipment has aged, only a minor correction causes system trouble.

- Data simplicity: since the data in the network is for the purpose of control, commands are simple and sequential.
- Real-time processing: difficulty of inserting alarms for security to minimize the response time.
- Linkage with information systems (intra-network): planning an integration of networks for the streamlining of management.
- Generalization of equipment: Linux and Windows have begun to be installed on the equipment and the TCP/IP protocol is being used.
- Ubiquitous user: the ubiquitous web evolves around the expectations of users who want to interact with information and services from anywhere.
- The ubiquitous web: the web delivers and integrates information, services, and user data.
- The ubiquitous user agent: running on a wide range of devices such as desktop computers.
- Botnets opportunistically scan the internet to attack poorly configured or absence of security patches.
- Zero-day exploit: updated software and the newest security patches may still have vulnerabilities.
- The insider attack: employees with access to the system.
- Errors in new software products.

Due to the above vulnerabilities, there may be security threats [Chiesa2009, Pollet2002]. Particularly:

- (1) Possibility of an intrusion incident when ICS is linked to an enterprise system.

  The situation is similar in the banks, in securities firms and in insurance companies. Private IP is used to protect the transaction systems of a bank, the asset management system of a securities firm, the customer management server and the servers for the management of accounting and production information and cables are used for safe communication. There have been many intrusion incidents by insiders. For instance, when customers use the internet for banking or making inquiries about their insurance policy or the production information though a Web server, the internal systems may be under attack through the servers being exposed to the outside. The internal control system and main systems may be under attack by viruses and hacking through partially exposed Management Information Systems (MIS) servers or PCs. Even though Intrusion Detection Systems (IDS) or firewall or virus vaccine programs can protect the unknown attacks, they are of no use in blocking new intrusion methods and patterns, so we are still exposed to a high risk of vulnerabilities.
- (2) Possibility of remote intrusion into the control systems using utilities and tools.

When attacks are made remotely using utilities and tools made to be connected with the programs developed for the operation of ICS/SCADA, such systems may lose control power. As is shown in intrusion cases, when utilities are used to directly control the control systems, it would be hard for IDS and firewalls to detect and block the intrusions and also difficult for the operators to notice them. When wireless terminals are used to check and control the servers, as they are generally used nowadays, the servers are regarded as exposed to the risk outside. Remote control functions using cell phones will be more evolved and sophisticated with additional services while the risk of intrusions will increase if the utilities and tools installed in the wireless terminals should be taken by someone outside.

- (3) Intrusion by the vendors of the control systems due to the connection of services or ports for remote access and support.

  There is a possibility of including backdoor and Trojan horse software. When connected through a specific port, and a manager has the power to control the communication with a specific service, there is a possibility of intrusion. After the control system has been deployed and delivered, sometimes vendor access for remote maintenance remains enabled. In this case, intrusion incidents may occur.

- (4) Possibility of intrusion when trying to control the control system by insiders using a remote management tool.

  Nowadays, servers are managed after placing them in IDC (Internet Data Center) or a certain secure location. Most managers do not work in front of their systems and instead, they use remote management tools to manage and control their systems. Most companies manage their servers after placing them in IDC and the trend will continue. PC Anyware, Terminal Server and VNC (Virtual Network Computing) are some of the most used remote management tools. When SCADA and DCS systems are controlled by a remote management tool, the target system for remote access usually can make a detour of ACL (Access Control List) network implemented two or three fold and can be the subject of a direct attack. The target system can also be expected to be effectively used as a means to avoid the intrusion detection and the intrusion blocking system.

Another list of possible vulnerabilities of ICS is provided by the US Homeland security [Homeland2010]:

Published vulnerabilities:

- Use of vulnerable remote display protocols;
- Secure Shell daemons that allow older versions of the protocol and are vulnerable to a downgrade attack;
- Anti-virus and spyware programs that do not have current signatures or are updated in such a manner that open an attack vector;
- Lack of a patching process/schedule leaves the ICS hosts open to attack from publicly disclosed vulnerabilities;

- Domain hosts using or storing antiquated LAN Manager (LanMan) hashes, which can be cracked using a dictionary attack;
- Backup software vulnerabilities that allow the attacker to manipulate data or server.

Web vulnerabilities:

- Web HMI vulnerabilities;
- Secure Sockets Layer man-in-the-middle attacks where the attacker takes advantage of self-signed HyperText Transfer Protocol over Secure Socket Layer (HTTPS) certificates.

Input validation vulnerabilities:

- Buffer overflows in ICS services;
- Structured Query Language (SQL) injection.

Improper authentication:

- Authentication bypass, e.g. client-side authentication;
- Use of standard Information technology (IT) protocols with clear-text authentication;
- Cyber Security Assessments of Industrial Control Systems Good Practice Guide 24
- Unprotected transport of ICS application credentials.

Improper access controls (authorisation):

- Wireless LAN access that can be used to get to the control network;
- Blank system administrator password on a Microsoft SQL Server database, which allows remote administrator access to the database and the server itself;
- VPN configuration problems that unintentionally allow clients unfettered access to the corporate, DMZ, or control LAN;
- System management software that allows central management of multiple servers may allow an attacker easy access to the same hosts;
- Common processes (any process that is installed and listening on multiple boxes), which if compromised, provide access to multiple hosts;
- Weak firewall rules;
- Circumvented firewalls;
- Shared printers that span security zones. This may provide a network transition that does not traverse the firewall;
- Unsecure network device management.

ICS data and command message manipulation and injection;
Database vulnerabilities;
Unnecessary or risky services and applications:

- Internet/e-mail access from within secure zones (DMZ, SCADA) may allow malware inside these protected zones.

3.4 Attack Vectors

3.4.1 Cyber attack sources

Threats to SCADA systems may arise from two different sources, mainly internal employees and external attackers. The threat from internal employees is real but not very likely as it would be easier to identify the attacker in most cases and the fear of the consequences would in itself reduce the likelihood of such attacks. On the other hand, it is easier for an external attacker to launch cyber attacks and the attack could go undetected, thereby making the SCADA systems more vulnerable.

Essentially two basic sources of attacks can be distinguished [Minich]:

1. Internal

- Non malicious: employees or contractors causing unintentional damage
- Malicious: system users with extensive internal knowledge of the system who intentionally cause damage

2. External

- Opportunistic: hackers seeking a challenge
- Deliberate: malicious, well-funded political activists, organized crime, or nation states

According to the above classification, following there are related examples of historical attacks [Minich]

Internal/Non-malicious: On June 10, 1999, a pipeline owned by Olympic Pipeline Company ruptured causing gasoline to leak into two creeks in Bellingham, Washington. The gasoline ignited, resulting in a fireball that killed three people, injured eight others, and caused significant property damage. It released approximately ¼ million gallons of gasoline to the environment. Although external pipeline damage, improperly installed pressure relief valves, and a failure of the controllers of the SCADA system were the clear culprits, it was the lack of policies and procedures at the Olympic Pipeline Company that led to this catastrophe. Evidence points to operator errors due to inadequate access controls and audit policies, and no security training.

Internal/Malicious: The Maroochy Water Services cyber attack incident of April 2000 is a good example of an insider attack on an industrial SCADA system. Vitek Boden worked for the Hunter Watertech firm that installed radio-controlled SCADA equipment for the Maroochy Shire Council in Queensland, Australia. Boden left his job at Hunter Watertech and applied for a job with the Maroochy Shire Council, but was turned down. Boden later proceeded to hack into the Maroochy Water Services SCADA system through the radio communications network using a radio and laptop computer. He used his knowledge and experience with the SCADA system to issue commands, disable alarms, and manipulate data through the local controllers to hide problems from the from the system's central monitoring computers. His tampering resulted in 800,000 litres of raw sewage spills. Maroochy's lack of access control policies and procedures for their system was the main cause of this incident. Additionally, the lack of an incident response plan, security training, and audit policies did not help to mitigate the attack or the effects afterwards.

External/Opportunistic see Directed *attacks and Advanced Persistent Threats*, further discussed in section 3.6

3.4.2 Targeted attacks

Targeted Cyber Attack Types: Malicious attackers can launch targeted attacks such as sniffing packets at an Internet service provider (ISP) or carrier and then maliciously modifying the packets in the network to achieve the expected results. They could proactively exploit software bugs and other vulnerabilities in various systems, either in the corporate network or the SCADA network, to gain unauthorized access to places such as control center networks, SCADA systems, interconnections, and access links. Openly available vendor documentation for proprietary CI (i.e. power systems) control software also makes them vulnerable to software exploits. They could configure unauthorized access points to send false information to confuse the SCADA systems in order to trigger unwanted countermeasures. They could target RTUs, Intelligent Electronic Devices (IEDs), uplink connections, and other physical entities to disrupt services. They could exploit the deterministic nature of the inter-center control communications protocol (ICCP) messaging protocol to achieve the desired effects on the SCADA network and the CI (i.e. electric grid).

Flood-based Cyber Attack Types: Cyber-attacks that are based on denial of service (DoS) mechanisms, and others that spread due to viruses and worms by causing a traffic avalanche in short durations, can potentially bring down systems and cause a disruption of services. There is no well-known, fool-proof, defence against such cyber attacks in the computing literature. Various effective ad-hoc solutions have been adopted on traditional computer networks. If the access links that connect the SCADA network to the Internet are swamped by heavy traffic caused by such attacks, it could prove disastrous as the control and supervisory data (including alarms, IED data) flowing to the SCADA network could be lost in the network. The gateway or firewalls installed to monitor the incoming traffic could be overloaded by the large volumes of attack traffic. Thus the ability of the SCADA network to respond to actual failures can be significantly affected. Also, the traffic flood could contain malicious ICCP messages that could confuse the SCADA systems to a great extent. There are many other avenues through which an attacker can execute a cyber attack in a manner that allows the attack to go undetected. Well-known techniques in computing literature, e.g., source address spoofing, or domain name system (DNS) cache poisoning, could also be tried but the impact of these attacks is currently unknown and needs further study.

3.4.3 SCADA communication protocols

Some of the potentials attacks harming a SCADA system are performed through communication stack by using the TCP/IP or the Internet reference. In particular, those attacks involve different layers like the network, transport and application layer or the implementation of protocols.

In the following we report some attacks that involve the network layer:

1. Diagnostic Server Attacks through UDP port. Adversaries have access to the same debugging tools that any Real-Time Operating System (RTOS) developers do. For example, the RTOS VxWorks debug service that runs UDP on port 17185 is enabled by default thus an attacker can execute the following attacks without any authentication.

2. Idle Scan: is to blind port scan by bouncing off a dumb “zombie” host, often a preparation for attack. Both MODBUS and DNP3 have scan functionalities prone to such attacks when they are encapsulated for running over TCP/IP.

3. Smurf: is a type of address spoofing that is implemented by sending a continuous stream of modified ICMP packets to the target network with the sending address that is identical to one of the target computer addresses. In the context of SCADA systems, if a PLC acts on the modified message, it may either crash or dangerously send out wrong commands to actuators.

4. ARP Spoofing/Poisoning: The Address Resolution Protocol (ARP) is primarily used to translate IP addresses to Ethernet MAC addresses and to discover other connected interfaced device on the LAN. The ARP spoofing attack is to modify the cached address pair information. By sending fake ARP messages which contain false MAC addresses in SCADA systems, an adversary can confuse network devices, such as network switches. When these frames are falsely sent to another node, packets can be sniffed; or intentionally to a host connected to different actuators, then physical disasters of different scales are initiated. Static Machine Access Control Address (MAC address) is one of the counter measures. However, certain network switches do not allow static setting for a pair of MAC and IP address. Segmentation of the network may also be a method to alleviate the problem in that such attacks can only take place within same subnet.

5. Chain/Loop Attack: In a chain attack, there is a chain of connection through many nodes as the adversary moves across multiple nodes to hide his origin and identity. In case of a loop attack, the chain of connections is in a loop make it even harder to track down his origin in a wide SCADA system.

Regarding the attacks that involve the transport layer, SCADA protocols, particularly those running over top of transport protocols such as TCP/IP have vulnerabilities that could be exploited by attacker through methodologies as simple as injecting malformed packets to cause the receiving device to respond or communicate in inappropriate ways and result in the operator losing complete view or control of the control device.

A representative example is the SYN flood which is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic. A mitigation strategy for SYN flood attacks on SCADA systems is described in [Grimes2005] and it is based on client puzzles that force clients, including attackers, to use computational resources to calculate the solution to a cryptographic puzzle or hash function. Once the client returns a valid solution, the connection is completed and data exchange begins.

Moving on the application layer, it is important to remark that currently there is no strong security control in protocols used in SCADA systems. Practically there is no authentication on source and data such that for those who have access to a device through a SCADA protocol, they can often read and write as well. The write access and diagnostic functions of these protocols are particular vulnerable to cyber and cyber induced physical attacks. Next, we list potential attacks associated with more SCADA specific protocols:

1. DNS forgery: sends a fake DNS reply with a matching source IP, destination port, request ID, but with an attacker manipulated information inside, so that this fake reply may be processed by the client before the real reply is received from the real DNS server.

2. MODBUS: the lack of encryption or any other security measures of MODBUS exposes this protocol to different vulnerabilities which have been analyzed in [Triangle2007]. One of them - force Single and Multiple Coils - is to manipulate a MODBUS frame by changing the function code in order to switch off remote devices and suppress output thus to create a false sense of situation at the HMI side. That implies that attacks can include DoS (e.g., rebooting Modbus servers) reconnaissance (e.g., unauthorized reading of data, and gathering device information), and unauthorized write requests.

3. DNP3: due to its lack of security, it suffers from the same weaknesses of MODBUS.

In the following, some attacks on implementation of protocols are presented:

1. TCP/IP: protocols implementation in Windows based machines exhibit some vulnerabilities that be exploited in machines that do not have up-to-dated patches. An example is the DoS attack named WinNuke which sends a string of OOB (out of band) data to the target computer via a TCP segment causing it to crash. That may not damage or change the data on the computer hard disk, but any unsaved data would be lost and the machine should be restarted.

2. OPC (Object Linking and Embedding for Process Control): is a series of standard specifications for use in process control and manufacturing automation applications to facilitate interoperability between software applications and process hardware. These protocols present different vulnerabilities. An example is the opportunistic DoS attack [CERT2006] that installs a malware on a machine of the company network which begins to search for OPC targets. When it detects any OPC servers on the

control system, it can attack any vulnerable applications using the OPC vulnerabilities. Once this scenario occurs, the OPC server will be unavailable and may require anything from a simple reboot to complete software re-installation and configuration to recover.

3. ICCP: is a protocol used by utility organizations throughout the world to provide data exchange over WANs among utility control centers, utilities, power pools, regional control centers. LiveData ICCP Server [Nai2007] implementation of the International Organization for Standardization (ISO) Transport Service over TCP exhibits a heap-based buffer overflow that allows an attacker to trigger the overflow to execute arbitrary code or crash a LiveData ICCP Server to cause a DoS attack.

3.5 Security incidents – indirect attacks and other issues

This section describes security incidents that involved ICS infrastructures, albeit not targeting them directly.

3.5.1 CSX Train Signalling Systems

In August 2003, a computer virus managed to infect critical systems on the United States of America east coast train-signalling infrastructure at CSX Corp.'s, shutting down the signalling and dispatching systems for 23 states east of the Mississippi. Several trips were cancelled and trains were delayed between15 minutes and 6 hours [Niland2003].

The cause was pinpointed to a Sobig virus infection [Nahorney2003]. This virus propagates from host to host using infected attachments in e-mail messages, rapidly spreads to other victims, using the address book from the first victim. It also creates a backdoor that can be used by a malicious hacker to gain control of a computer or to upload spambot applications.

3.5.2 Zotob

In August 2005, the Zotob Work [Roberts2005] crashed thirteen of DaimlerChrysler's United States. automobile manufacturing plants forcing them to a downtime of almost an hour. Zotob is a worm that spreads by exploring a buffer overflow vulnerability in the Windows operating system – it affects computers by slowing them down and causing them to continually crash and reboot.

While the Zotob worm itself does not carry a destructive payload, it leaves an open backdoor control channel that allows attackers to control the infected machine. The worm also adds several lines of code into a machine to prevent it from accessing certain antivirus websites. Zotob and its variations also caused computer outages at Caterpillar Inc., Boeing, and several U.S. news organizations.

3.5.3 Slammer

In January 2003, the Slammer worm infected a private computer at the idled Davis-Besse nuclear power plant in Ohio, disabling a safety monitoring system for almost five hours. Also, the plant process computer failed, taking about six hours to recover [Poulsen2003]..

The Slammer worm explores a Microsoft SQL Server vulnerability without any destructive payload – when it infects a machine, it generates random IP addresses to contact and replicate other systems. Having entered on the nuclear power plant through an unsecured contractor network, the worm managed to cross to the ICT network through a T1 line that bypassed the corporate firewall. Once in the ICT network, the worm managed to spread to the ICS network, infecting an unpatched Windows server and affecting communications on the control networks of five other utilities, at such a pace that control traffic was blocked (the work is very small, only 376 bytes – something that eases propagation).

3.5.4 Hatch Nuclear Power Plant Shutdown

In March 2008, the Hatch Nuclear Power Plant in Georgia went through an emergency shutdown as a result of a software update on the plant's ICT network [Aalto2008], which incidentally, had two-way communication with the ICS network. Reset after a reboot: the SCADA safety systems detected a lack of data and signalled that the water level in the cooling systems for the nuclear fuel rods where below acceptable levels, causing an automatic shutdown.

While engineers were aware of the two-way communication link (across a firewall), they didn't know that there was a possibility of an update to propagate to the ICS network. While there was no danger to the public, the power company lost millions of dollars in revenue. In the aftermath, the engineers chose to close all physical connections between the SCADA and business networks, to avoid further problems.

3.6 Security incidents – directed attacks and advanced persistent threats

This section describes attacks and security incidents that specifically targeted SCADA systems. These attacks can be also classified as Advanced persistent threats (APT).

APT usually refers to a group, such as a foreign government, with both the capability and the intent to persistently and effectively target a specific entity. The term is commonly used to refer to cyber threats, in particular that of Internet-enabled espionage, but applies equally to other threats such as that of traditional espionage or attack. Other recognised attack vectors include infected media, supply chain compromise, and social engineering. Individuals, such as an individual hacker, are not usually referred to as an APT as they rarely have the resources to be both advanced and persistent even if they are intent on gaining access to, or attacking, a specific target.

The global landscape of APTs from all sources is sometimes referred to in the singular as "the" APT, as are references to the actor behind a specific incident or series of incidents. The Stuxnet computer worm has been described by one Middle East Consultant as "state terrorism". In this example, the Iranian government might consider the Stuxnet creators to be an Advanced Persistent Threat.

3.6.1 Maroochy Shire Sewage Spill

On January 2000, the Maroochy Shire Counci's sewage control system in Queensland, Australia was targeted by an attack, almost immediately after the control system for the sewage plant was installed by a contractor company [Slay2007]. The plant suffered a series of problems, with pumps failing to start or stop and alarm events not being reported. Also, there was intermittent loss of communications between the control center and the pumping stations.

At first, the system operators suspected of a pipe leak, but later they found that valves were being open without being commanded to do so, but the hypothesis of an attack was not considered. Only after months of logging they found spoofed controllers activating the valves – later on, they discovered the cause: an ex-employee of the contractor that originally had installed the system was trying to convince the water treatment company to hire him to solve the problems they were having.

The result of this attack was the flooding of the grounds of nearby locations (one hotel, park, and river) with approximately 264,000 gallons of sewage. This kind of cyber-attacks may be difficult to detect – in fact, the response was slow enough that the attacker was able to perform 46 documented attacks before being caught.

3.6.2 Stuxnet

Unlike most malware, Stuxnet [O'Murchu2011] does little harm to computers and networks that to not meet specific configuration requirements. While the worm is promiscuous, it makes itself inert if Siemens software is not found on infected computers, and contains safeguards to prevent each infected computer from spreading the worm to more than three others, and to erase itself on 24 June 2012. For its targets, Stuxnet contains, among other things, code for a man-in-the-middle attack that fakes industrial process control sensor signals so an infected system does not shut down due to abnormal behaviour. Such complexity is very unusual for malware. The worm consists of a layered attack against three different systems [O'Murchu2011]:

1. The Windows Operating System
2. Siemens PCS 7, WinCC and STEP7 industrial software application that run on Windows
3. One or more Siemens S7 PLCs

Stuxnet attacked Windows systems using an unprecedented four zero-day attacks (plus the CPLINK vulnerability and a vulnerability used by the Conficker worm). It is initially spread using infected removable drives such as USB flash drives, and then uses other exploits and techniques such as peer to peer RPC (peer-to-peer systems with remote procedure call) to infect and update other computers inside private networks that are not directly connected to the Internet. The number of zero-day Windows exploits used is unusual, as they are valued, and crackers do not normally waste the use of four different ones in the same worm. Stuxnet is unusually large at half a megabyte in size, and written in several different programming languages (including C and C++) which is also irregular for malware.

The Windows component of the malware is promiscuous in that it spreads relatively quickly and indiscriminately. The malware has both user-mode and kernel-mode rootkit capability under Windows, and its device drivers have been digitally signed with the private keys of two certificates that were stolen from separate companies, JMicron and Realtek, which are both located at Hsinchu Science Park in Taiwan. The driver signing helped it install kernel-mode rootkit drivers successfully and therefore remain undetected for a relatively long period of time. Both compromised certificates have been revoked by VeriSign. Two websites in Denmark and Malaysia were configured as command and control servers for the malware, allowing it to be updated, and for industrial espionage be conducted by uploading information. Both of these websites have subsequently been taken down as part of a global effort to disable the malware.

According to German researcher Ralph Langner, once installed on a Windows system Stuxnet infects project files belonging to Siemens WinCC/PCS 7 SCADA control software, and subverts a key communication library of WinCC called s7otbxdx.dll. Doing so intercepts communications between the WinCC software running under Windows and the target Siemens PLC devices that the software is able to configure and program when the two are connected via a data cable. In this way, the malware is able to install itself on PLC devices unnoticed, and subsequently to mask its presence from WinCC if the control software attempts to read an infected block of memory from the PLC system. The malware furthermore used a zero-day exploit in the WinCC/SCADA database software in the form of a hard-coded database password.

The entirety of the Stuxnet code has not yet been understood, but its payload targets only those SCADA configurations that meet criteria that it is programmed to identify. Stuxnet requires specific slave variable-frequency drives (frequency converter drives) to be attached to the targeted Siemens S7-300 system and its associated modules. It only attacks those PLC systems with variable frequency drives from two specific vendors: Vacon based in Finland and Fararo Paya based in Iran. Furthermore, it monitors the frequency of the attached motors, and only attacks systems that spin between 807 Hz and 1210 Hz. The industrial applications of motors with these parameters are diverse, and may include pumps or gas centrifuges. Stuxnet installs malware into memory block DB890 of the PLC that monitors the Profibus messaging bus of the system. When certain criteria are met, it periodically modifies the frequency to 1410 Hz and then to 2 Hz and then to 1064 Hz, and thus affects the operation of the connected motors by changing their rotational speed. It also installs a rootkit-the first such documented case on this platform-that hides the malware on the system and masks the changes in rotational speed from monitoring systems.

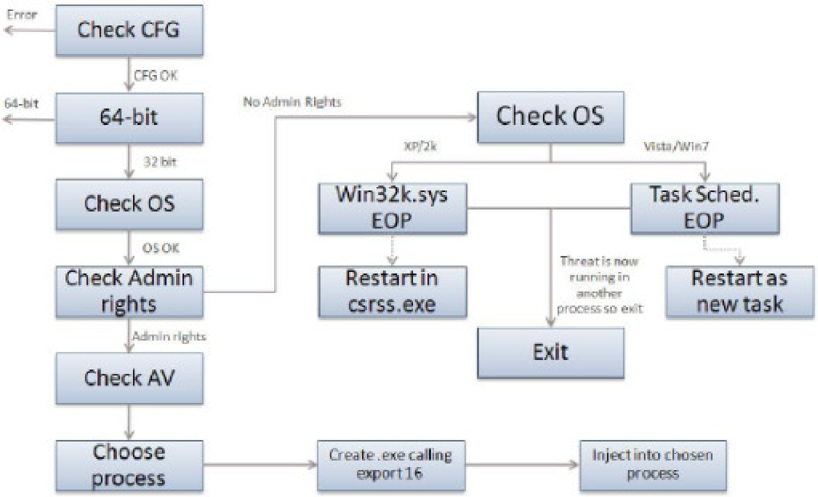


Figure 3-8: How Stuxnet works [O'Murchu2011]

Figure 3-8 shows the steps that Stuxnet made to penetrate the system. Once it reach a computer, it makes some checks, like the architecture (Stuxnet works only on 32-bit system with Windows XP/2k or Vista/Win7), so it checks if it has the Admin right and got a specific procedures for every OS that is suited for. Than it checks for the antivirus and chooses a new process to infect.

3.6.3 DuQu

On October 14, 2011 Symantec was alerted to a sample by a research lab with strong international connections that appeared very similar to the Stuxnet worm [Symantec2011]. The threat was recovered from an organization based in Europe. They have confirmed Duqu is a threat nearly identical to Stuxnet, but with a completely different purpose. The threat was written by the same authors, or those that have access to the Stuxnet source code, and appears to have been created after the last Stuxnet file they recovered. Duqu's purpose is to gather intelligence data and assets from entities such as industrial control system manufacturers in order to more easily conduct a future attack against another third party. The attackers are looking for information such as design documents that could help them mount a future attack on an industrial control facility.

Duqu does not contain any code related to industrial control systems and is primarily a Remote Access Trojan (RAT). The threat does not self-replicate. The telemetry shows the threat has been highly targeted toward a limited number of organizations for their specific assets. The attackers use Duqu to install another info-stealer that can record keystrokes and collect other system information. Duqu consists of a driver file, a Dynamic Link Library (DLL) (that contains many embedded files), and a configuration file. These files must be installed by another executable (the installer) which has not yet been recovered. The installer registers the driver file as a service so it starts at system initialization. The driver then injects the main DLL into services.exe. From here, the main DLL begins extracting other components and these components are injected into other processes. Duqu uses HTTP (HyperText Transfer Protocol) and HTTPS to communicate to a command and control (C&C) server at 206. [REMOVED].97, which is hosted in India. Through the command and control server, the attackers were able to download additional executables, including an info-stealer that can perform actions such as enumerating the network, recording keystrokes, and gathering system information. The information is logged to a lightly encrypted and compressed local file, and then must be exfiltrated out.

The threat is configured to run for 36 days. After 36 days, the threat will automatically remove itself from the system. Duqu shares a great deal of code with Stuxnet; however, the payload is completely different. Instead of a payload designed to sabotage an industrial control system, it has been replaced with general remote access capabilities. The creators of Duqu had access to the source code of Stuxnet, not just the Stuxnet binaries. The attackers intend to use this capability to gather intelligence from a private entity that may aid future attacks on a third party [Symantec2011].

3.6.4 Night Dragon

Night dragon (ND) is an attack that was developed in the recent years. ND involve social engineering, spear-phishing attacks, exploitation of Microsoft Windows OS vulnerabilities, Microsoft Active Directory compromises, and the use of RAT in targeting and harvesting sensitive competitive proprietary operations and project financing information with regard to oil and gas fields bids and operations [McAfee2011].

Detail of the attack

Attackers using several locations in China have leveraged C&C servers on purchased hosted services in the United States (US) and compromised servers in the Netherlands to wage attacks against global oil, gas and petrochemical companies, as well as individuals and executives in Kazakhstan, Taiwan, Greece and the US to acquire proprietary and highly confidential information. The primary operational technique used by the attackers comprised a variety of hacker tools, including privately developed and customized RAT tools that provided complete remote administration capabilities to the attacker. RATs provide function similar to Citrix or Microsoft Windows Terminal Services, allowing a remote individual to completely control the affected system. To deploy these tools, attackers first compromised perimeter security controls, through SQL-injection exploit of extranet web servers, as well as targeted spear-phishing attacks of mobile worker laptops, and compromising corporate VPN accounts to penetrate the targeted company’s defensive architectures and conduct reconnaissance of targeted companies’ networked computers.

SQL injection attacks

1. Attacker craft a HTTP GET request to inject commands to SQL server to gain system-level access
2. Malware is placed on server and used to harvest the local and Active Directory account credentials
3. Active Directory accounts are used to access network that connects with remote C&C address
4. Attacker uses RAT malware to conduct additional reconnaissance and systems compromises and to harvest confidential data

Spear phishing attacks

1. Attacker sends a spear-phishing email containing a link to a compromised server
2. User opens infected email and the compromised website is accessed; a RAT is downloaded
3. User account information and host configuration information is sent to a C&C server
4. Attacker uses RAT malware to conduct additional reconnaissance and systems compromises and to harvest confidential data

3.6.5 Common phase among Stuxnet, DuQu, Night Dragon and others

In all the attacks, the first goal is to infect a computer. Duqu is pretty similar to Stuxnet, so the technique used to penetrate a computer is the same. Stuxnet and Night Dragon have similar approach to the first infection. They trust that an employee make an error (like to plug in an untrusted Universal Serial Bus (USB) drive or click on link on a email that redirect to a fake page). When the employee makes the wrong action, a malware is installed on his / her computer. That malware can hide itself from the antivirus (if any) and doesn't infect too many computers, because doing that they are exposed to a great traffic on the net. When a computer is compromised, they use one (or more) vulnerabilities (known only by the attacker or known and fixed by the distributor but don't applied by the end-user) to hide and spread itself. Installing a backdoor they can communicate to an external server that can upgrade the malware with new instruction or simply collect sensible data.

Stuxnet does not do anything until all the devices are infected, so if any is healed, it is immediately re-infected. Stuxnet and Night Dragon are attacks that are accomplished from the inside.

An attack always came from the inside. An attack accomplished directly by the attacker is the one of Australia using the WiFi. In that case social engineering is the way to follow for gain the right. This kind of menace is potentially the most destructive because the attacker knows well the system and then, he can act undisturbed in the best way (for him). There are even tools for brute forcing a password, hacking the fingerprint's access system or even tools that reveal common misconfiguration of router or any device (even SCADA device) [Chiesa2009].

3.6.6 Exploitation of the Aurora vulnerability in SCADA test bench

In 2007, the US Department of Energy's Idaho laboratory performed a cyber-attack test on an industrial sandbox including a Diesel generator controlled by a SCADA system. The test was successful and burst into the national spotlight in September 2007, when CNN reported the result with a video showing the destruction of the generator [CNN 2007]. The picture (on the right) shows that the generator was made to break down. The information was considered so sensitive that the Department of Homeland Security made a request to CNN to not divulge certain details about the experiment. Indeed, it is worrying that coordinated attacks, based on this experiment, could cause prolonged outages in large sections of the electrical grid in the USA.

Note that the experiment was so successful that some experts considered the video to be a hoax. The US Department of Defense was required to address in an unclassified document (for public distribution) a summary of the experiment and an official statement regarding the authenticity of the experiment results: "*During Aurora's initial discovery and validation in 2007, the issue attained extremely high visibility, which eventually led to interest from the National Security and the Homeland Security Advisor to the President. […] At some senior government levels Aurora has been incorrectly briefed as a computer virus, other ranking officials have been told that a simple software patch will fix the problem. As the result, decision makers have been denied a fair opportunity to make accurate and responsible risk management decisions based on the fact". [DOD2009].*

The diagram in Figure 3-9 presents a description of the test bench used to perform the Aurora attack.

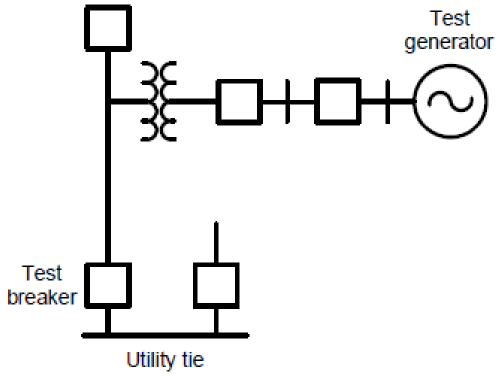


Figure 3-9: Aurora attack test bench.

What was the goal of the Aurora attack? As mentioned in the analysis by Schweitzer Engineering Laboratories [Sal2009]; "*the intent of the Aurora attack is to intentionally open a breaker and close it out of synchronism to cause damage to the connected generators and motors*" [SAL2009]. If the attacker succeeds to open and close it out of synchronism, he could provoke a resonance phenomenon which would lead to generator outage (and partial destruction). Indeed, the open-close sequence induces an increase in the speed and torque at generator level which are well managed in normal mode but not during a cyber-attack (cf. Figure 3-10: Relationship of torque, speed and breaker status

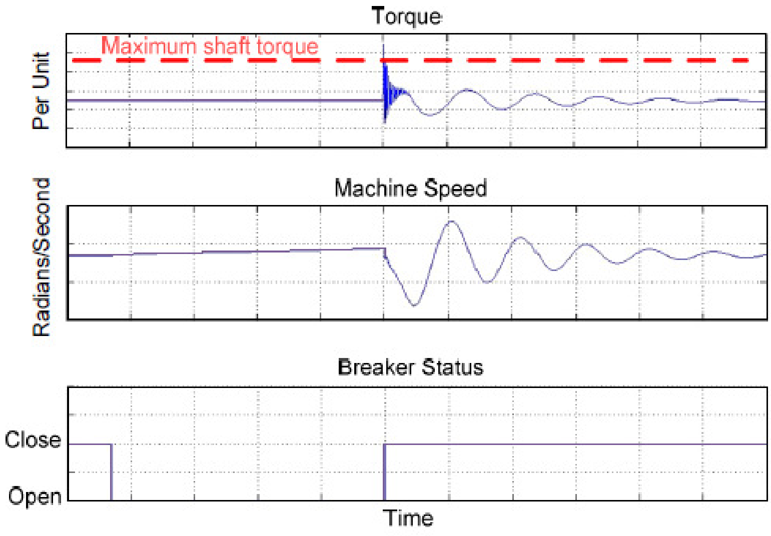


Figure 3-10: Relationship of torque, speed and breaker status

Even more interesting in the paper by Schweitzer Laboratory is the root cause analysis of the successful Aurora attacks. Excluding the countermeasures, which could be applied on the industrial hardware in terms of machine design or protection design, this analysis shows that:

- common security measures such as security management good practices described in ISO 27002 (strong password, incident management on SCADA) or common ICT technical countermeasures (encryption, authentication mechanism) could mitigate the Aurora risk (cf. above)
- security is often based on very small details which separately seem non-threatening (because during normal functioning everything goes well), but combined give a very good opportunity for an attacker to perform malicious actions.

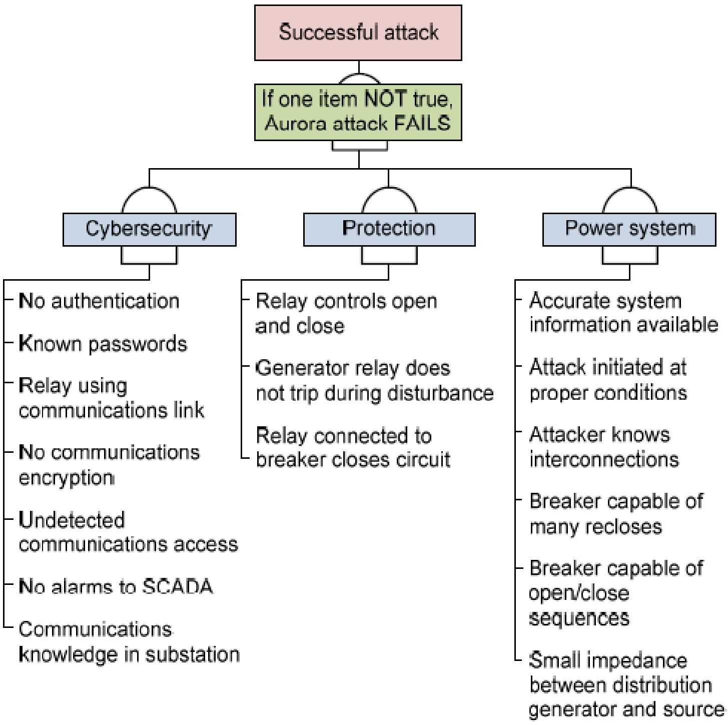

Successful attack
If one item NOT true, Aurora attack FAILS
Cybersecurity
Protection
Power system
No authentication
Known passwords
Relay using communications link
No communications encryption
Undetected communications access
No alarms to SCADA
Communications knowledge in substation
Relay controls open and close
Generator relay does not trip during disturbance
Relay connected to breaker closes circuit
Accurate system information available
Attack initiated at proper conditions
Attacker knows interconnections
Breaker capable of many recloses
Breaker capable of open/close sequences
Small impedance between distribution generator and source

Figure 3-11: Root cause analysis of Aurora attack

The different components of Aurora attack could be described as follow:

Table 3-1: Aurora attack components

ID	Parent	Name	Attack Vector	Operational Impact	Defense	Informational Impact	Target
001	-	Aurora	Social Engineering	User Compromise (know password)	Awareness	Disclosure	User
001	-	Aurora	Design flaw	User compromise (no authentication or weak authentication mechanism)	Awareness	Discovery	Local
002	001	Aurora	Design flaw	Timeliness degradation (action on reachable breaker)	Shielding (encryption of communication)	Distort	Local
002	001	Aurora	Social Engineering	Misuses of resources (communication channel hacked)		Disclosure	Network protocol

003	002	Aurora	Incorrect permission	Denial of service/Destruction	Awareness Replacement	Distort	Local
003	002	Aurora	Design flaw (no alarm)	Denial of service/Destruction	Shielding Replacement		Local

4 Related Work

This chapter describes a number of European projects regarding CI security and resilience. Moreover, it gives an overview of some standards regarding security and safety in CI environments.

4.1 European projects

4.1.1 ESCoRTS

The ESCoRTS project (European network for the Security of Control and Real-Time Systems) was an European Union (EU) Framework Programme 7 (FP7) Project started in 2008. It evolved leading manufacturers of control equipment, EU process industries and research institutes, to foster progress towards cyber security of industrial control systems in Europe. Its key objectives included developing a common understanding of industrial requirements regarding the security of control systems and the related standardisation, accompanied by an awareness-raising program reaching all stakeholders. Its final objective was to assist the EU as a whole (i.e. authorities, industry, manufacturers, etc.) in developing informed positions and in shaping current and future efforts related to control systems security standardisation.

The ESCoRTS project had the main objective of increasing security in control systems, through the dissemination and use of good and recognized practices applied to the field, together with the creation and adoption of standards [ESCoRTS]. It includes several partners, including equipment manufacturers (ABB, Areva, Siemens) and customers (Ente Nazionale per l'energia ELettrica (ENEL), Transelectrica, Mediterranea delle Acque).

The work is centered on the application of standardization and normalization on the control system level, not on the evaluation of its efficacy. In terms of security for control systems, the project decided for the adoption of the most promising and broad-scope standard for the purpose, the ISA99/IEC62443 [ESCoRTS2010]. The project evaluated several other alternatives (a total of 37 standards – 14 from the USA, 10 European and 13 International).

ESCoRTS also addressed the topic of security metrics for cyber security assessment [ESCoRTS2010b], in an effort to deal with the lack of legal requirements or ceritfications for security in industrial control applications. It also proposed some metrics that could provide the basis for more complete and specific developments.

A study on the requirements for laboratories for security research on industrial control systems was also performed by ESCoRTS [ESCoRTS2011], resulting in a set of valuable guidelines for building security research testbeds.

The project also produced attack and vulnerability taxonomies, grouped into 4 main categories:

- **Architectural**: related to design issues (such as deficient network infrastructure planning and deployment) or lack of isolation between process, control and ICT networks.
- **Security policy-related:** related to lack of software lifecycle management policies (updates), access control policies, non-repudiation mechanisms, badly maintained or absent documentation and security auditing practices.
- **Software:** security bugs, lack of update and patching.
- **Communication protocols:** related with vulnerabilities in communication protocols used in control networks.

Also, several categories of attacks were analysed (protocol-oriented, process-oriented, exchange network targeted), as well as several countermeasures to reduce their impact.

4.1.2 INSPIRE

INSPIRE (Increasing Security and Protection through Infrastructure REsilience) [INSPIRE] was a FP7 project, started in 2008, with the main objective of increasing the security and resilience of infrastructure control systems by means of a self-reconfigurable architecture suitable for SCADA systems.

The INSPIRE project effort went along several different action lines:

- Analysis and modelling of vulnerabilities of networked process control systems.
- Design and implementation of techniques and architectures for increasing security and resilience of networked controls systems.
- Verification, validation and integration of the developed tools.
- Exploitation, dissemination and standardization.

To increase the resilience of such systems INSPIRE proposed to develop traffic engineering algorithms, self-reconfigurable architectures and diagnosis and recovery techniques in order to protect critical information infrastructures by appropriately configuring, managing, and securing the communication networks which interconnected the distributed control systems

4.1.3 AFTER

AFTER (A Framework for electrical power sysTems vulnerability identification, dEfense and Restoration) [AFTER] is a EU FP7 project started in September 2011 that addresses the challenges posed by the need for vulnerability evaluation and contingency planning of the energy grids and energy plants considering also the relevant ICT systems used in protection and control.

Project emphasis is on cascading events that can cause catastrophic outages of the electric power systems. The main addressed problems are related to high impact wide spread multiple contingencies, the most significant wide area criticality. This kind of contingencies and the following cascading effects can be caused by deliberate acts of terrorism, sabotage, criminal activity, malicious behaviour or they can simply be caused by a combination of accidents, natural disasters, and negligence. Both risk analysis and risk mitigation will be pursued. In particular, two major objectives are addressed.

The first is to develop a methodology and tool for the integrated, global vulnerability analysis and risk assessment of the interconnected Electrical Power Systems considering their interdependencies. This objective meets the TSO (Transmission System Operator) need to overcome current approaches based on separate evaluations of either power system or ICT system. Further, the adoption of risk concepts allows a more in depth, quantitative evaluation of the security of the electrical power system.

The second objective is to develop algorithms and tools supporting contingency planning in a two-fold approach: (a) preventing or limiting system disruption, by means of physical security techniques and defence plans; (b) re-establishing the system after a major disruption, by means of restoration plans. To this aim, AFTER propose the use of the global risk assessment methodologies as a support to defence plan design. A language to model defence plans functionalities and ICT architecture is developed. New defence plan concepts are also introduced to cope with emergency situations.

4.1.4 CRISALIS

CRISALIS (CRitical Infrastructure Security AnaLysIS) [CRISALIS] is a EU FP7 project aims at providing new means to secure critical infrastructure environments from targeted attacks, carried out by resourceful and motivated individuals. The recent discovery of the Stuxnet malware shows that these threats are already a reality. Their success in infiltrating Critical Infrastructure environments is calling attention on the ineffectiveness of standard security mechanisms at detecting them. Stuxnet is believed to have been operating undetected for almost one year leveraging multiple vulnerabilities that were previously unknown, and has been discovered only as a consequence to an operational anomaly that triggered the attention of the field operators. This fact clearly shows that our methods to find vulnerabilities and detect ongoing or successful attacks in critical infrastructure environments are not sufficient.

CRISALIS focuses on these two aspects: detection of vulnerabilities and attacks in critical infrastructure environments. We address two different, yet interlinked, use cases that are typical for the power grid infrastructure: control systems based on SCADA protocols and the Advanced Metering Infrastructure. CRISALIS leverages the unique characteristics of critical infrastructure environments to produce novel practical mechanisms and techniques for their security assessment and protection. This is achieved by pursuing three main research objectives: (i) providing new methodologies and techniques to secure critical infrastructure systems; (ii) providing new tools to detect intrusions; (iii) developing new, more effective, techniques to analyze infected systems. Particular attention is paid to ensure the practical implementation of these techniques in real-world environments, and to minimize the impact on operations, goals which are attainable thanks to the direct involvement in the process of end users and device manufacturers who provide expertise and realistic test environments to validate the proposed methodologies. CRISALIS partners include Symantec, Alliander, Chalmers Technical University, ENEL, Eurecom, SecurityMatters, Siemens, and the University of Twente.

4.1.5 PRECYSE

PRECYSE (Prevention, protection and REaction to CYber attackS to critical infrastructures) [PRECYSE] is an EU project that will define, develop and validate a methodology, an architecture and a set of technologies and tools to improve -by design- the security, reliability and resilience of the ICT systems supporting the Critical Infrastructures (CI).

The proposed solutions will be validated in two demonstrations in the domains of transport and energy. All the process will be strongly user-driven, with not only two high profile user organisations forming part of PRECYSE consortium, but also a powerful User Group which spans through multiple application domains –energy, transport, defence and police forces, utilities, public authorities, etc.-and all European regions, from Southern Europe to Scandinavia.

4.1.6 SAFEGUARD

The project formally started on 1st December 2001, although work on the project did not fully start until January 2002, and finished on 31st May 2004.

Safeguard's aim was to enhance the dependability and survivability of Large Complex Critical Infrastructures (LCCIs), such as distributed electric and telecommunication networks. Modern automation systems underlying LCCIs include different levels of automation, regulation, and control, but "intelligent" functions relating to critical issues such as system dependability and survivability are usually monitored or executed by human operators. Safeguard can improve the dependability and survivability of large infrastructures as perceived by all interested parties: the owners, operators and customers. The main objective of the project was to provide a systemic conceptual framework and an integrated software toolkit that, employed within an intelligent multi-agent system, enhances the dependability and survivability of Large Complex Critical Infrastructures (LCCIs).

Safeguard set up two test beds within two domains: telecom and electricity.

4.1.6.1 Telecom test bed

The test network at Swisscom currently consists of over 100 machines, including routers and switches in three different sub networks. The test network is made up of two zones (server / work) subject to attacks and one Safeguard zone especially protected for the Safeguard and maintenance system. A wide variety of operating systems with different patch systems are available, e.g. Windows, LINUX, HP Unix, BSD, Solaris 2.6-2.9. The whole test network is reachable from the World Wide Web (WWW) via a jump station; thus partners can develop and test their agents in a real environment. Fault/attack scenarios were based on standard attack tools and attack scripts. These scenarios include attacks of all kinds, as well as worms and DoS. In order to generate reproducible scenarios, Swisscom investigated realistic, but simple failures, misconfigurations and attacks, which actually happen every day in the real environment but never get detected in time (due to the lack of Safeguard). One result of running Safeguard in the test bed has been that it is clear that Safeguard's functionality improves after constant operation in the test network; this is especially true for the anomaly detection.

4.1.6.2 Electricity test bed

The test bed at Italian National Agency for New Technologies, Energy and Sustainable Economic Development (ENEA) consists of a SCADA emulation environment made up of five machines that provide a Control Centre, some data concentrator devices connected with RTUs, the platform containing the Safeguard agents, and a console from which it is possible to design, generate and run faults and malicious attack scenarios. In its final version, the test bed works using an IEEE 24 bus electricity network that is used by electricity engineers for tests and experiments on these types of networks. The utilisation of such a network required improvements in the capability of the emulated SCADA environment. An additional requirement for the test bed is also the possibility to use a local version of the e-AGORA Simulator. 'Fault and attack' has been defined in terms of fault/attack goals, phases and sequences of actions. Some 'generic attack scenarios' have been defined, for which the principal tools/methods utilised by hackers to violate/monitor/corrupt the operating system SCADA environment are utilised. A fault/attack scenario tool is utilised to produce and run the fault and attack scenarios in a more formal way. It also gives the possibility of logging the attack/fault action sequences and more easily documenting the results of the tests. Preliminary tests were executed to study the behaviour of single low level agents. More complex tests, aimed at activating the reaction of the whole Safeguard system involving low and high level agents, were executed for a range of scenarios.

4.1.7 VIKING

VIKING (Vital Infrastructure, Networks, Information and Control Systems Management) [VIKING] is a FP7 Project (36 months) started in November 2008. The main objectives of VIKING are: (a) to investigate the vulnerability of SCADA systems and the cost of cyber attacks on society; (b) to propose and test strategies and technologies to mitigate these weaknesses; (c) to increase the awareness for the importance of critical infrastructures and the need to protect them.

Society is increasingly dependent on the proper functioning of the electric power system, which in turn supports most other critical infrastructures: water and sewage systems; telecommunications, internet and computing services; air traffic, railroads and other transportation. Many of these other infrastructures are able to operate without power for shorter periods of time, but larger power outages may be difficult and time consuming to restore. Such outages might thus lead to situations of non-functioning societies with devastating economical and humanitarian consequences. For this reason, this consortium has decided to concentrate its research to the systems for transmission and distribution of electric power. We anticipate that most of the results will be applicable to the protection of other critical infrastructures.

The operation and management of the electric power system depend on computerized industrial control systems. Keeping these systems secure and resilient to external attacks as well as to internal operational errors is thus vital for uninterrupted service. However, this is challenging since the control systems are extremely complex. Yet, the systems are operating under stringent requirements on availability and performance: If control and supervision are not done in real-time, the power network may come to a collapse.

These computerized control system, normally called SCADA standing for System for Control And Data Acquisition, includes functions for remote acquisition of vast amounts of data from measurements placed in strategic points, e.g. power stations, in the geographically widely spread electrical and for the remote control of process devices. Many SCADA systems include computerized models of the process which enables simulation of alternatives process states and of optimization. Due to legal and environmental constraints, e.g. for building of new high voltage power lines or power stations, the primary process itself is difficult to expand which in its turn leads to higher and higher utilization of the existing transmission and generation resources. The process is, in other words, operated closer to its physical limits. Those the SCADA systems are becoming increasingly critical for the operation of the process and therefore are becoming a critical component for the availability and security of the supervised infrastructure.

The objective of the VIKING project is to develop, test and evaluate methodologies for the analysis, design and operation of resilient and secure industrial control systems for critical infrastructures. Methodologies will be developed with a particular focus on increased robustness of the control system. As mentioned, the focus is on power transmission and distribution networks. The project combines a holistic management perspective—in order to counteract sub-optimization in the design—with in-depth analysis and development of security solutions adapted to the specific requirements of networked control systems.

The traditional approach to verify the security of SCADA systems has been ad-hoc testing of existing commercial SCADA system in laboratory environments. The systems to be examined have been installed in different labs and tested by skilful people searching for cyber attacks vulnerabilities. The focus in these tests has been on the protection of the central computer system of the SCADA system, since the central computer system has most connections to the external environment through office networks and Internet.

In the VIKING project we will take an alternative and complementary approach to SCADA system security. Firstly we will study the whole control system from the measurement points in the process itself over the communication network to the central computer system as illustrated in the following picture with the yellow exclamation marks indicating potential targets for cyber attacks.

4.1.7.1 State Estimator-based Attacks and Mitigations

One of the interesting issues of the Viking project are state estimator–based attacks and mitigations.

In Figure 4-1, a schematic block diagram of a modern power network control system is shown. Note that the figure presents a very simplified picture of these complex systems, and only components explicitly treated in this part of VIKING are included.

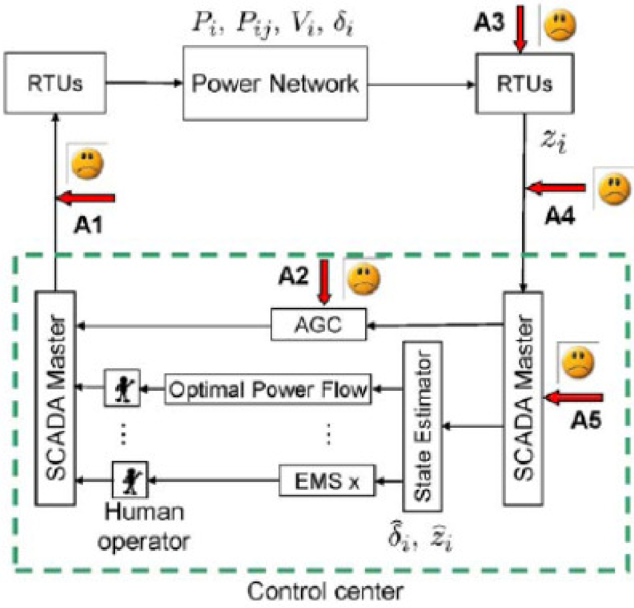


Figure 4-1: Block diagram of power network, SCADA and control center [VIKING]

The considered power network models are on the transmission level. They should be thought of as large and consisting of up to hundreds of buses that are spread out over a large geographic area (a region in a country, for example). VIKING country network, which consists of 40 buses has been considered. To monitor and control the behaviour of such large-scale systems, SCADA systems are used to transmit measurements, status information, and circuit-breaker signals to and from Remote Terminal Units (RTUs) that are connected to substations.

Today a modern SCADA system is supported by Energy Management Systems (EMSes) such as automatic generation control (AGC), optimal power flow analysis, and contingency analysis (CA). For such large-scale systems, lost data, failing sensors, or lack of sensors in certain areas, are common. The incoming data is therefore often fed to a so-called state estimator (SE) which provides EMS and the human operator in the control center with hopefully accurate information at all times. For example, the SE will provide estimates of power flows and injections that are not even measured. To remove faulty data possibly due to noise, the state estimator is supplemented by its Bad Data Detection (BDD) system. The

BDD system works by checking that the received measurement data reasonably well matches a physical model of the power network. However, as SCADA/EMS systems are increasingly more connected to office LANs in the control center, these critical infrastructure systems are potentially accessible from the internet. The SCADA communication network is also heterogeneous and consists of fibre optics, satellite, and microwave connections. Data is often sent without encryption. Therefore many potential security threats exist for SCADA/EMS systems. In particular, Viking has studied how an attacker can inject false data at points A3-A5 in Figure 4-1, while avoiding triggering the BDD system. This means the state estimator will provide false state information to the human operator, while he/she does no warnings. Hence, the human operator could be fooled to for instance destabilize the system or to run it in a non-optimal operating point. The data attack A3 could be conducted by an attacker that hijacks an RTU in the field to transmit false data. The data attack A4 could be conducted by an attacker that intercepts the communication going to the control center. Finally, the attack A5 could be conducted by an attacker that accesses the database in the SCADA master system.

Security indices were introduced, that measure how hard it is to perform undetectable false-data attacks against the SE, as described above. One index measures ”attack hardness” by counting the minimum number of sensors that needs to be corrupted together with the target sensor to avoid detection by the BDD system.

Also the feasibility of the data attacks against the SE in the VIKING country 40-bus system has been verified by conducting experiments in the VIKING test bed. A framework to analyze and study the impact of a class of stealthy deception attacks targeting the SE component through measurement data corruption has been provided.

4.1.7.2 ViSiCi - Cities Simulator

This tool attempts to simulate a virtual society [VIKING2010], in a simplified, albeit functional way. This society has the typical characteristics and properties of a modern, normal civilization, incorporating dynamic and static structures. It also includes basic infrastructures like buildings, streets and electric utility service networks, and even public and private organizations producing goods and services to consumers.

This simulated society was denominated as the VIKING country, being implemented as a template, which was extended and is available for every single European country, tailored with specific the demographic, economic (Eurostat) and energy provider statistics of each one. This society relates energy needs with economic activity, incorporating the energy demands based on the Gross Domestic Product (GDP) of the simulated country.

This simulator evaluates the impact of energy supply failure scenarios, estimating the monetary and non-monetary impact on the society. The nature of such failures is simplified – when they occur it is supposed for all economic activity to stop and there is no production or consumption of goods until the power supply is re-established.

Re-establishment of energy supply is modelled in a more complex way. Depending on the activity, there might be an increased demand for electricity during the restart of the activity – for instance, the recovery of a refrigeration system implies spending more energy to reach the configured temperatures depending on the length of the interruption.

The evaluation of costs and consequences from such interruptions considers two dimensions:

- Monetary: cost is calculated based on the difference between GDP with and without the failure.
- Non-monetary: macro and micro perspectives (see Figure 4-2) of the consequences on the society, evaluated accordingly with the duration of the failure and its incidence.

The micro perspective is focused on the evaluation from the individuals' standpoint. It considers aspects such as the loss of property or transportation issues, lack of fuel, among others. It is evaluated accordingly with the duration of the failure event. The micro perspective considers the consequences in line with the number of affected individuals, evaluating the probability of mutinies, or epidemic eruptions, just to mention some examples.

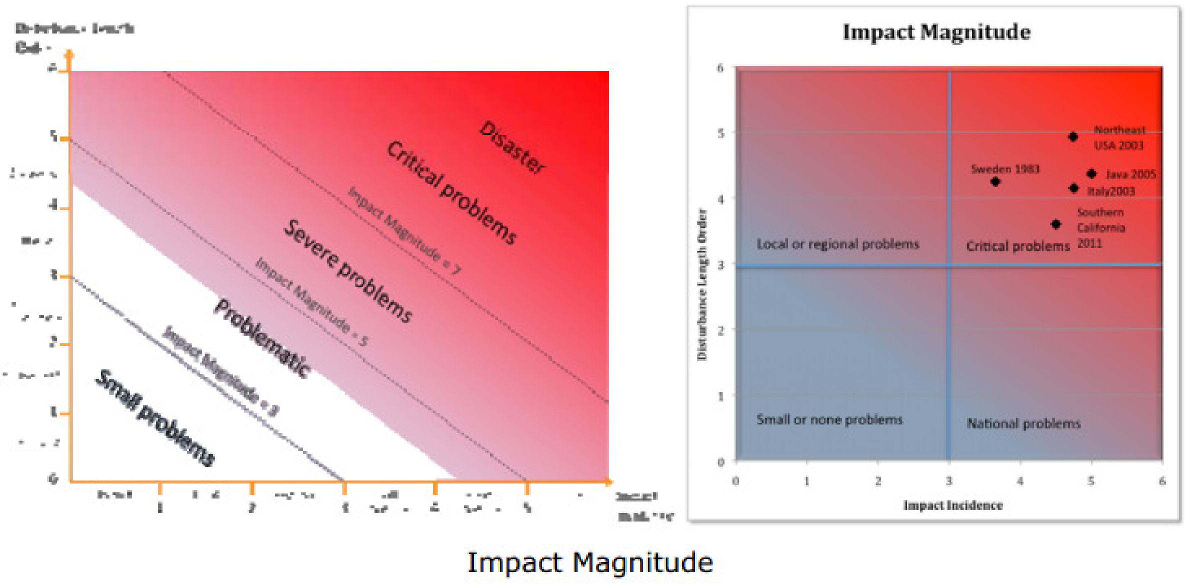


Figure 4-2: Evaluation of impact on society from the micro and macro perspective ([VIKING2010a])

4.1.7.3 CySeMoL: a Cyber security modelling language

CySeMoL is a language (or Meta model) in which system architectures are described. The language contains general-purpose entities such as services, data flows, operating systems, as well as security specific entities such as intrusion detection systems, firewalls and patch management processes. The language also defines how these concepts can be related to each other as well as some important properties (from a security perspective) of the entities, such as for instance if an operating system is using non-executable memory or if services have known vulnerabilities.

With the language, users of CySeMoL are able to describe their system architectures. In addition to this purely descriptive part of CySeMoL, a mechanism for calculating a value that could be considered a security index is also included in the language. In essence, this mechanism is an attack/defense graph, which describes how different attacks and attack steps could be performed in the system architecture and its different components. So, depending on the exact configuration of the architecture, different attack paths will be possible for an attacker to accomplish. For all those attack paths, CySeMoL provides numerical estimates for how likely it is that all the different attack steps are possible to accomplish. These estimates are given as conditional probabilities (specified in Bayesian networks).

Data for the CySeMoL calculation mechanism

At the core of the CySeMoL lie the conditional probabilities used for the calculations. These figures have to a large extent been collected by asking security experts in surveys on their opinions of the impact of different countermeasures on different attacks, such as the DoS example above. For all questions the explicitly stated assumption to the respondent has been that the attacker is a professional penetration tester with one week of preparation. Some of the figures are also deterministically derived, and some have been derived from previously published studies. In total four surveys has been conducted on various parts of the CySeMoL with answers from 165 respondents as maximum and a handful of respondents as minimum. In order to identify qualified respondents (identifying which experts that really are experts) Cooke’s classical method has been used. This method essentially weight different respondents depending on how good they are at answering some test questions relevant to the area of the survey questions (that the CySeMoL developers have known the answers to). This means that only a few of the for instance 165 respondents mentioned above performed good enough to be called experts. A philosophical note worth mentioning is that Cooke’s method tries to identify the true answer to questions rather than to have a large amount of answers to generalize from. If the truth is found it does not matter how many respondents that have stated it. All answers, i.e. conditional probabilities, have been collected also including the respondents’ opinion on the uncertainty of the answer (expressed as a three point estimation).

Table 4-1: Conditional probabilities of an attack [VIKING2010a]

Does de software have vulnerabilities?	Does the attacker have access credentials?	Probability of a successful attack
Yes	Yes	0.72
Yes	No	0.53
No	Yes	0.60
No	No	0.38

For instance, for the first estimate in Table 4-1 above the average answer (of the respondents selected as the “true experts”) is 72%. However, there is a 5% chance that the value is below 32% (again, on average) and a 95% chance that the value is below 95% (on average). Another way of expressing it would be that there is a 90% chance that the attack success value is between 32% and 95%. As we can see from this example the 72% figure here is acquainted with a large share of uncertainty. However, the calculations made in the present version of CySeMoL, only the expected mean value is used, i.e. 72%, in the example above. Intended usage of CySeMoL The intended usage of the CySeMoL is to support security analyses of SCADA and control system architectures. It should support users that are not necessarily security experts themselves. If the user provides a system architecture, the CySeMoL can provide a security estimate in terms of attack probabilities. So, by analysing different architectures and different attack processes the user can get a better understanding of available weak spots in the architecture. In addition, it also provides a clue on how effective different mitigation strategies (probably) are. As described above, the figures provided are often acquainted with quite big uncertainty. This imposes that the calculated percentage value figure should be treated with care. The results should be seen as a support for reasoning about different alternative scenarios or mitigations. On average a scenario with attack success probability of 10% is more resilient towards the analysed attack process than one with 30%, even included the uncertainties (that in general are the same magnitude for scenarios). Essentially the user needs to define two things: 1) the system architecture (including a number of properties), and 2) which targets they would like to analyse as well as starting points for the attacks.

CySeMoL delivers results for (the most probable attack path between) pairs of a single starting point and a single target. But, in order to have a more complete and holistic understanding of the whole architecture several such pairs needs to be considered.

Again, comparing scenarios without analysing the complete set of potential pairs will provide an indication of their relative security.

Since the CySeMoL is quite large and complex, it is extremely time consuming to do the calculations by hand. Thus, all examples in the project have used the Enterprise

Architecture Analysis Tool (EAAT) to calculate the results and visualize the models. The EAAT as such is however not developed within the VIKING project.

4.2 ICS cyber security standards and initiatives

The North American Electric Reliability Corporation (NERC) [NERC] has constituted the compliance standard CIP 1200 for a power system to meet the network security requirements [Chee-Wooi2007]. This standard provides general guide lines about what to comply and alert, and training of the personnel.

The guidance includes identification of physical and cyber parameters, and critical cyber asset; however, it does not provide system vulnerability assessment based on what is implemented. Some other SCADA security standards are available, e.g., BS7799 by British Standard Institute (BSI), IEC/ISO 17799, ISA TR 99.00.02, AGA12 by American Gas Association, and 21 steps by Department of Energy. Some of these standards provide guidance that include domain specific defences with examples [Torkilseng2006].

As far as standards as concerned, the Common Criteria (CC) also known as ISO/IEC 15408 [ISO15048] is the most recognised standard in the area of security evaluation ad assurance. The CC describes a framework in which developers can specify their security requirements and testing laboratories can evaluate the products to determine if they actually meet the claimed security. The CC also permits comparability between the results of independent security evaluations. It does so by providing a common set of requirements for the security functions of IT products and systems and for assurance measures applied to them during a security evaluation. In fact Part 3 of the CC defines the assurance requirements both for the development environment and for the product itself as well as the tasks for the evaluator. These assurance requirements are organised in classes, then in families of components, which include functional specification and design descriptions, testing, lifecycle management, delivery procedures, security of the development environment, and vulnerability analysis. Developers can either build up their own consistent assurance package or use one of the seven predefined Evaluation Assurance Levels (EAL). EAL1 to EAL7 provide an increasing scale that balances the level of assurance obtained on the product security with the cost and feasibility of acquiring that degree of assurance. Unfortunately applicability of the CC is restricted to end products and thus cannot be entirely used to address the complexity of operational systems. This is due to the fact that the evaluated entity in CC is considered to be relatively stable, within a closed region, separated from the surrounding environment with a predefined set of threats addressed within a protection profile [Hecker2009].

ISO/IEC TR 19791 (ISO/IEC, 2006b) provides guidance and criteria for the security evaluation of operational systems. It provides an extension to the scope of ISO/IEC 15408 by taking into consideration a number of critical aspects of operational systems not addressed in the ISO/IEC 15408 evaluation. The principal extensions address evaluation of the operational environment surrounding the target of evaluation, and the decomposition of complex operational systems into security domains that can be separately evaluated.

4.2.1 NERC

NERC Standards CIP-002-3 through CIP-009-3 provides a cyber security framework for the identification and protection of Critical Cyber Assets to support reliable operation of the Bulk Electric System. These standards recognize the differing roles of each entity in the operation of the Bulk Electric System, the criticality and vulnerability of the assets needed to manage Bulk Electric System reliability, and the risks to which they are exposed.

Business and operational demands for managing and maintaining a reliable Bulk Electric System increasingly rely on Cyber Assets supporting critical reliability functions and processes to communicate with each other, across functions and organizations, for services and data. This results in increased risks to these Cyber Assets. Standard CIP-002-3 requires the identification and documentation of the Critical Cyber Assets associated with the Critical Assets that support the reliable operation of the Bulk Electric System. These Critical Assets are to be identified through the application of a risk-based assessment.

4.2.2 NIST

The NIST provides recommendation to implement Industrial Control Systems and to handle Security Incidents:

NIST. Keith Stouffer, Joe Falco, Karen Scarfone, "*Guide to Industrial Control System (ICS) security*" (Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC)), NIST Special Publication 800-82, June 2011.

NIST. Peter Mell, Karen Kent, and Joseph Nusbaum, "*Guide to Malware Incident Prevention and Handling*", NIST Special Publication 800-83, November 2005.

4.2.3 HOMELAND SECURITY

Homeland security recommended practice: improving industrial control systems cybersecurity with defense-in-depth strategies. The Department of Homeland Security (DHS) supported institutes such as NIST and laboratories to ensure an high level of security awareness on Critical Infrastructure Security (CIP) and Critical Information Infrastructure Protection (CIIP).

"Critical infrastructure protection report," Critical Infrastructure Protection GAO-05-434, Department of Homeland Security Faces Challenge in Fulfilling Cybersecurity Responsibilities, May 2005.

4.2.4 ISO

4.2.4.1 ISO 270xx

The family of standard 270xx deals with assessment of information system security in the enterprise. The figure below gives an overview of the ISO 270xx framework.

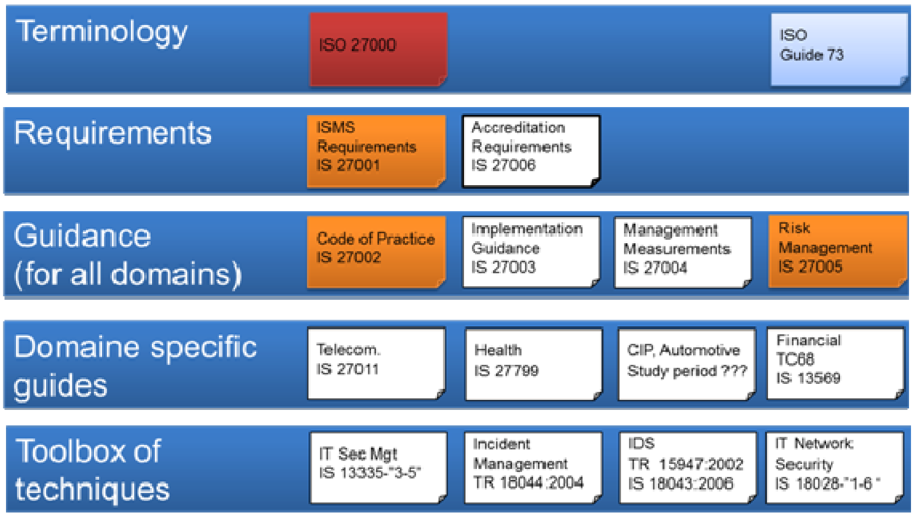

Terminology
ISO 27000
ISO Guide 73
Requirements
ISMS Requirements IS 27001
Accreditation Requirements IS 27006
Guidance (for all domains)
Code of Practice IS 27002
Implementation Guidance IS 27003
Management Measurements IS 27004
Risk Management IS 27005
Domaine specific guides
Telecom. IS 27011
Health IS 27799
CIP, Automotive Study period ???
Financial TC68 IS 13569
Toolbox of techniques
IT Sec Mgt IS 13335-"3-5"
Incident Management TR 18044:2004
IDS TR 15947:2002 IS 18043:2006
IT Network Security IS 18028-"1-6"

Figure 4-3: ISO 270xx framework

In the following, some of the most important standards are presented:

As per [ISO27001], "*ISO 27001 formally specifies a management system that is intended to bring information security under explicit management control. It requires that management: (a) systematically examine the organization's information security risks, taking account of the threats, vulnerabilities and impacts; (b) design and implement a coherent and comprehensive suite of information security controls and/or other forms of risk treatment to address those risks that are deemed unacceptable; (c) adopt an overarching management process to ensure that the information security controls continue to meet the organization's information security needs on an ongoing basis.*"

ISO 27002 is a guidance framework, which can serve as the basis for organizational risk assessment performance, also providing guidelines for security program planning..

ISO 27032 relates to cyber security, which is defined in the standard as the "*preservation of confidentiality, integrity and availability of information in the cyberspace*". It focuses on defining assets in the Cyberspace, threats, the role of stakeholders in cyber security and provides guidelines for stakeholders.

ISO 27033 relates to the security aspects involving the management, operation and use of information system networks.

4.2.4.2 ISO 18043

As per [ISO18043], "*ISO 18043 provides guidance for an organization that decides to include an intrusion detection capability within its IT infrastructure. It supports managers and users who want to: (a) understand the benefits and limitations of IDS; (b) develop a strategy and implementation plan for IDS; (c) integrate intrusion detection into the organization's* *security practices (d) understand the legal and privacy issues involved in the deployment of IDS.*

ISO 18043 provides information that will facilitate collaboration among organizations using IDS. The common framework it provides will help make it easier for organizations to exchange information about intrusions that cut across organizational boundaries.“

This standard gives an overview of the intrusion detection process, analysing the capabilities and restrictions of IDS systems, enumerating them. This is done to ease the selection of the best suited features for each specific case, also providing advice and guidance in several different aspects, from deployment to operational procedures for security alert management.

4.2.4.3 ISO 15408

As per [ISO15048], “*The standard ISO 15408, also called Common Criteria (CC), is a framework in which computer system users or/and manufacturers can specify their security functional and assurance requirements. According to these requirements, manufacturer can then implement and/or make claims about the security attributes of their products, and testing laboratories can evaluate the products to determine if they actually meet the claims. Users can implement products which reach a defined security level and increase the overall security level of their systems.”*

Pratically, the CC aims facilitating writing Security Targets (ST), i.e. a set of security features which has to be applied for a specific product (e.g. the X-Company #xxx Firewall) to reach a certain security level (Evaluation Assurance Level) (EAL). “*To allow consumer groups and communities of interest to express their security needs, and to facilitate writing Security Targets, the CC provides two special constructs: packages and Protection Profiles (PPs)*”. A PP is the description of a security profile applied to a type of product (e.g. any home firewall or business firewall) to reach the same security level. The aim of the PP or of a ST is to describe, according to the more rational manner as possible, the system and its environment, the threats which can impact the system, the required security objectives to reach the chosen security level, the Security Functional Requirements to reach these objectives and then to be able to define and to assess the technical solution implemented.

Why to use this type of approach? The following figure describes the advantage of this type of approach:

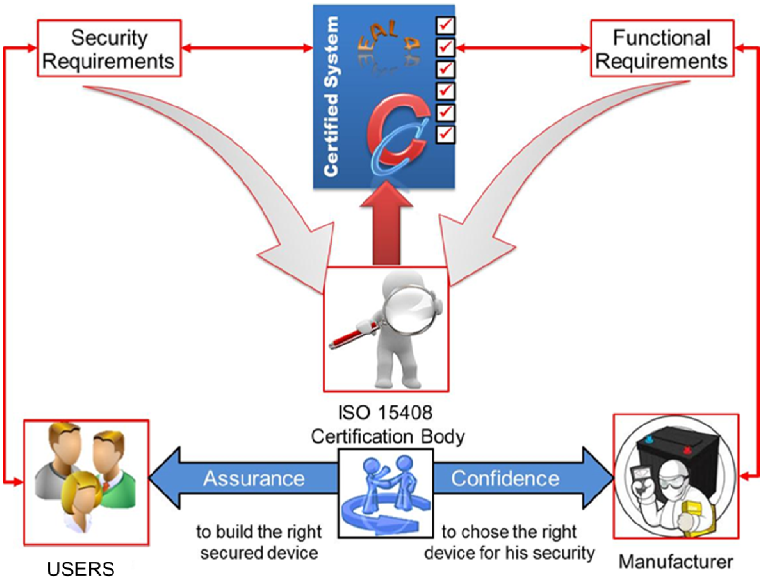


Figure 4-4: Common Criteria approach

In other words, it provides assurance that the process of specification, implementation and evaluation of a computer security product has been conducted in a rigorous and standard manner.

4.2.4.4 ISO 15443

As per [[ISO15443]], "*ISO 15443 describes a variety of IT security assurance methods and approaches and relates them to the IT security assurance framework in ISO/IEC TR 15443-1. The emphasis is to identify qualitative properties of the assurance methods and elements that contribute to assurance, and where possible, to define assurance ratings. This material is intended for IT security professionals for the understanding of how to obtain assurance in a given life-cycle stage of a product or service. The objective is to describe and categorize assurance methods and approaches in a manner enabling a review of their comparable and synergetic properties.*"

The basic idea behind this standard is to ease the selection of protection methods and mechanisms, eventually combining several different solutions in the scope of an IT security product, system, service and its particular environment.

4.2.4.5 ISO 19791

As per [ISO19791], "*ISO 19791 provides guidance and criteria for the security evaluation of operational systems. It provides an extension to the scope of ISO/IEC 15408, by taking into account a number of critical aspects of operational systems not addressed in ISO/IEC 15408 evaluation. The principal extensions that are required address evaluation of the operational environment surrounding the target of evaluation, and the decomposition of complex operational systems into security domains that can be separately evaluated. It provides: (a)* *a definition and model for operational systems; (b) a description of the extensions to ISO/IEC 15408 evaluation concepts needed to evaluate such operational systems; (c) a methodology and process for performing the security evaluation of operational systems; (d) additional security evaluation criteria to address those aspects of operational systems not covered by the ISO/IEC 15408 evaluation criteria.*"

This standard is constrained to security assessment of operation systems, not considering other scopes. Moreover, it does not define any techniques for enumeration, assessment/auditing or tolerance for operational risk.

4.2.5 ISA-99.00.02

The ISA-99 standard, provided by the ISA (International Society for Automation), is designed to be general in nature and can thus be applied to any of the critical infrastructure sectors. ISA99 is composed of two parts:

- Part I (ISA-99.00.01): *Security for Industrial Automation and Control Systems: Concepts, Terminology and Models*
- Part II (ISA-99.00.02): *Establishing an Industrial Automation and Control Systems Security Program*

Specific standards for SCADA security are provided by ISA-99.00.02, constituting a basic guidebook that an implementer of the ISA99 standard can use to assemble a security program, without prescribing the details for every industry.

The ISA99 standard defines specific security levels (levels 0-5) for control system and corporate IT components based on their function in the system. The architecture proposed by the ISA99 standard provides SCADA and control systems industry with a model for segmenting networks at levels 2 and above to ensure that SCADA and control systems are isolated from company data networks. The levels in ISA99 for a sample legacy SCADA/DCS Architecture are:

- Level 5 - Internet DMZ
- Level 4 - Enterprise IT includes
- Level 3 - Area Operations DMZ
- Level 2 - Area Supervisory Operations (HMI)
- Level 1 - Basic Process Control & Monitoring
- Level 0 - Instrumentations and Sensors

In order to improve ISA-99, the standard was superseeded by IEC 62443, which provides a more global approach to the security of industrial systems. It is similar to the ISO 2700x framework (Figure 4-5) and includes general guidelines (policies and procedures) and technical documentation (development requirements, technical security requirements etc…).

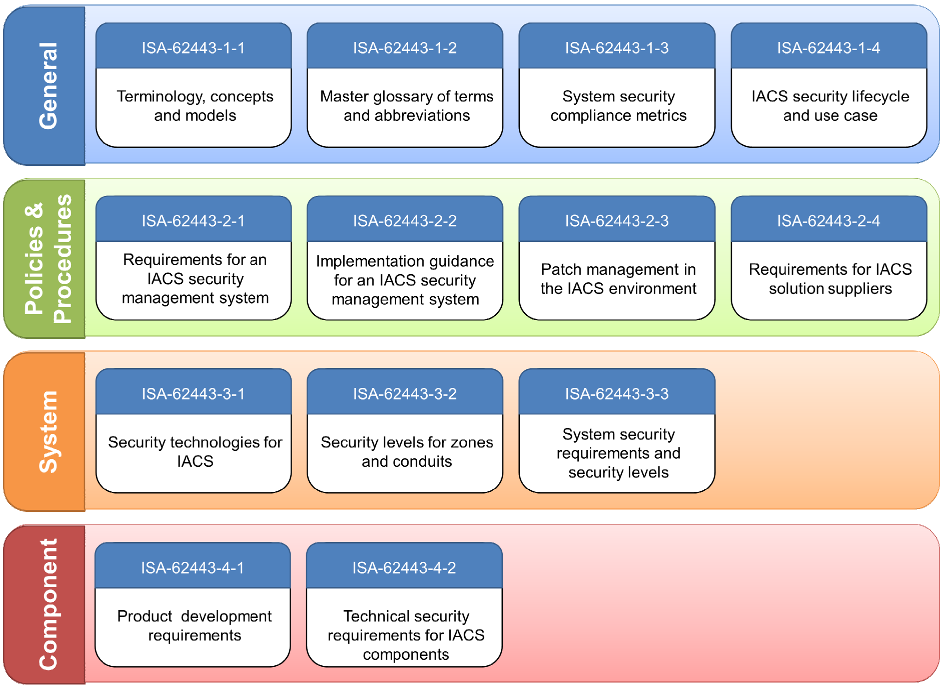

General
ISA-62443-1-1
Terminology, concepts and models
ISA-62443-1-2
Master glossary of terms and abbreviations
ISA-62443-1-3
System security compliance metrics
ISA-62443-1-4
IACS security lifecycle and use case
Policies & Procedures
ISA-62443-2-1
Requirements for an IACS security management system
ISA-62443-2-2
Implementation guidance for an IACS security management system
ISA-62443-2-3
Patch management in the IACS environment
ISA-62443-2-4
Requirements for IACS solution suppliers
System
ISA-62443-3-1
Security technologies for IACS
ISA-62443-3-2
Security levels for zones and conduits
ISA-62443-3-3
System security requirements and security levels
Component
ISA-62443-4-1
Product development requirements
ISA-62443-4-2
Technical security requirements for IACS components

Figure 4-5: ISA-62443 Standard Series (Adapted from [ISA99])

4.2.6 CSIS: 20 Critical Security Controls

The SANS institute has provided a short document to enhance the security level of sensitive infrastructure. Even if this document is not a formal standard, its recommendations become more and more applied by stakeholders to secure their infrastructure. The goal of the Critical Controls is *“is to protect critical assets, infrastructure, and information by strengthening your organization's defensive posture through continuous, automated protection and monitoring of your sensitive information technology infrastructure to reduce compromises, minimize the need for recovery efforts, and lower associated costs”.* [CCECD2013]

Description of Controls

Critical Control 1: Inventory of Authorized and Unauthorized Devices

Critical Control 2: Inventory of Authorized and Unauthorized Software

Critical Control 3: Secure Configurations for Hardware and Software on Mobile Devices, Laptops, Workstations, and Servers

Critical Control 4: Continuous Vulnerability Assessment and Remediation

Critical Control 5: Malware Defenses

Critical Control 6: Application Software Security

Critical Control 7: Wireless Device Control

Critical Control 8: Data Recovery Capability

Critical Control 9: Security Skills Assessment and Appropriate Training to Fill Gaps

Critical Control 10: Secure Configurations for Network Devices such as Firewalls, Routers, and Switches

Critical Control 11: Limitation and Control of Network Ports, Protocols, and Services

Critical Control 12: Controlled Use of Administrative Privileges

Critical Control 13: Boundary Defense

Critical Control 14: Maintenance, Monitoring, and Analysis of Audit Logs

Critical Control 15: Controlled Access Based on the Need to Know

Critical Control 16: Account Monitoring and Control

Critical Control 17: Data Loss Prevention

Critical Control 18: Incident Response and Management

Critical Control 19: Secure Network Engineering

Critical Control 20: Penetration Tests and Red Team Exercises

5 Reference Architecture for the Cyber Analysis and Detection Layer

This chapter describes the generic probing architecture proposed for the CockpitCI project in conceptual terms, as well as its components and fundamental concepts, with special relevance to the Intrusion Detection System (IDS) sensors and its respective placement.

5.1 Requirements for the cyber analysis and detection layer

CockpitCI attempts to leverage the work already developed In the MICIE project, by adding cyber-detection capabilities to get a broader perspective in terms of security, while reinforcing local decision-making capabilities, in case of high-risk, critical situations.

To achieve this purpose, there must be some kind of autonomy in terms of field devices (e.g. RTUs) in order to enable continuous and safe operation, even in the absence of control from a central SCADA control center. Thus, some kind of self-healing and protection capabilities must be incorporated at the RTU level, while safeguarding the basic principle that “everything that can be used, can be abused” – this means that this level of autonomy cannot be an obstacle to safe operation, potentially explored by an attacker or simply by interfering with the normal (and safe) operation of the ICS.

To overcome such contradictory behaviour a sort of hybrid schema will be considered and development in the project (from the CockpitCI Description of Work (DoW)):

- *at the level of Control Centre, the presence of an “Integrated On-line Risk Predictor” (a development of the one proposed by MICIE project) will perform an accurate situation assessment and will provide the operator with a qualitative/quantitative measurements of near future level of risk integrating data coming from the field, data coming from other infrastructures and data coming from smart detection agents monitoring possible cyber attacks.*
- *at field level, we complement the schema with a smart software layer for RTUs and a detection system for the TLC (Telecommunication) network. This layer will continuously analyze the inputs and outputs of the RTU in order to prevent misuse, and will analyze the traffic on the TLC network to recognize cyber attacks.*

This smart layer allows RTUs to retreat to a predefined “high risk” mode of operation were they might ignore SCADA commands, operating based on a “safe policy” behaviour profile. This is possible thanks to cyber-interdependencies modelling - according to the input of prediction tools (in case of coordinated attacks), it becomes possible to put in place a specific perimeter to detect potential coordinated cyber attacks on CIs for each type of detected attacks or for a mixed cyber attacks. As a safeguarding strategy, the previous defined perimeters can be used to isolate systems (such as RTUs) in case of too dangerous potential cyber-attacks, in order to check for tactical and operational solutions.

Therefore, the CockpitCI project includes a cyber analysis and detection layer (see Figure 5-1) that must work as a real-time (or, at least, as close as possible) Distributed Monitoring System and Perimeter Intrusion Detection System (PIDS). This PIDS must be able aggregate the filtered and analyzed information of potential cyber-attacks induced on SCADA systems or telecommunication involved in the operation of CIs, identifying the potential insecurities and vulnerabilities.

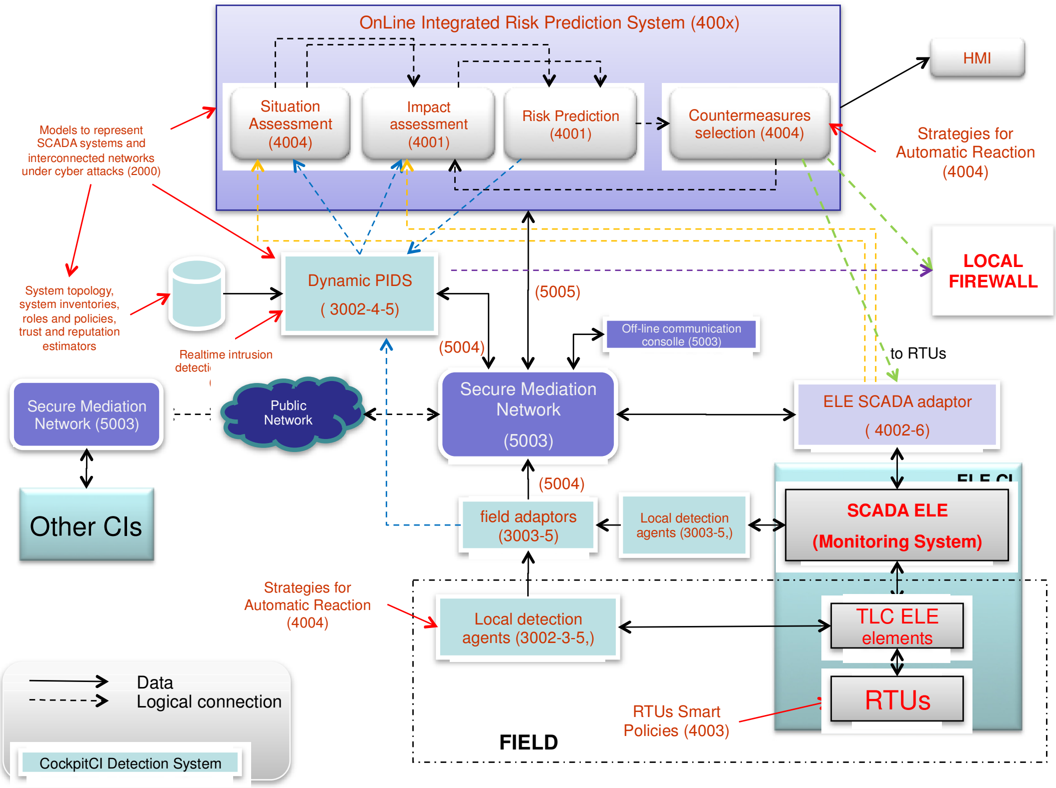

OnLine Integrated Risk Prediction System (400x)
Situation Assessment (4004)
Impact assessment (4001)
Risk Prediction (4001)
Countermeasures selection (4004)
HMI
Strategies for Automatic Reaction (4004)
Models to represent SCADA systems and interconnected networks under cyber attacks (2000)
LOCAL FIREWALL
System topology, system inventories, roles and policies, trust and reputation estimators
Dynamic PIDS
(3002-4-5)
(5005)
Off-line communication consolle (5003)
Realtime intrusion detecti
(5004)
to RTUs
Secure Mediation Network (5003)
Public Network
Secure Mediation Network
(5003)
ELE SCADA adaptor
(4002-6)
Other CIs
(5004)
ELE CI
field adaptors (3003-5)
Local detection agents (3003-5,)
SCADA ELE
(Monitoring System)
Strategies for Automatic Reaction (4004)
Local detection agents (3002-3-5,)
TLC ELE elements
RTUs
RTUs Smart Policies (4003)
Data
Logical connection
FIELD
CockpitCI Detection System

Figure 5-1: Overview of the Cyber-analysis and detection layer within the CockpitCI architecture

As such, the cyber-detection and Analysis layer hereby described (Figure 5-2) must be able to develop and deploy smart detection agents to monitor the potential cyber threats according to the types of networks (SCADA, IP…) and types of devices that belong to such networks, closely following the infrastructure classification discussed in section 2.1.

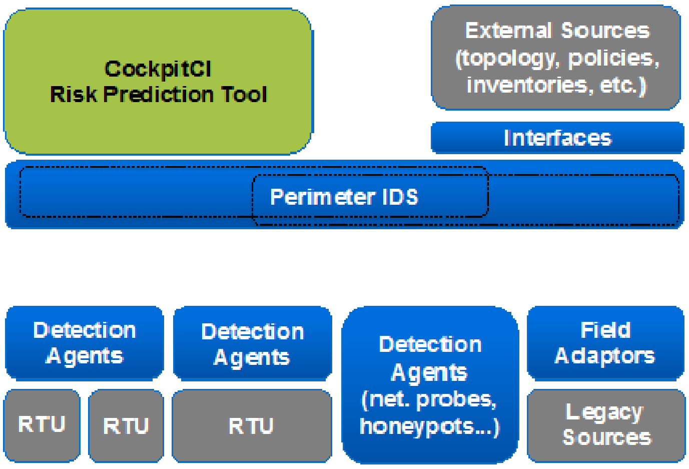


Figure 5-2: Simplified view of the CockpitCI Cyber-analysis and detection layer (blue blocks)

5.1.1 Functional requirements

In terms of functional requirements, the CockpitCI cyber-detection and analysis layer, must provide the following functionalities:

- Information from field adaptors must be taken into account, providing data from SCADA and Telco devices.
- It must encompass host, device and network-level detection mechanisms, such as Network Intrusion Detection Systems (NIDS), Host Intrusion Detection Systems (HIDS) or Hybrid IDS.
- Field-level mechanisms, such as RTU-level resources for operation monitoring and security detection must be provided. These mechanisms must be in line with the Smart RTU reaction system, in such a way that it must not interfere with its normal operation, providing complementary information.
- The architecture must include security detection and analysis as well as security auditing mechanisms.
- The solution must be able to accommodate different types of detection agents and security measures, accordingly with their deployment context.
- The system must interface with asset and inventory management mechanisms, in order to get a broad view of existing equipment, their function and location.
- Secure communications among all components must be enforced and be mandatory, both for data integrity and confidentiality.

5.1.2 Non-functional requirements

In terms of non-functional requirements, the CockpitCI cyber-detection and analysis layer, must provide the following functionalities:

- The CockpitCI cyber detection and analysis layer must be flexible enough to deal with the several event formats and involved reporting and alarming mechanisms.
- The CockpitCI cyber analysis and detection layer must be able to cope with several different detection techniques and tools, including conventional approaches (such as signature-based IDS tools and classic anomaly-based detection and event correlation) but also with more advanced solutions, particularly in two areas:
  - Adaptive machine learning based approaches, including innovative data mining and pattern recognition approaches towards event correlation, innovations in dynamic Bayesian networks, artificial neural networks, vector machines, fuzzy and evolutionary systems.
  - Aggressive usage of topology- and system-specific detection mechanisms based on the fact that the role and behaviour of each system component are expected to be more consistent over time than on other types of networks. Among other approaches, the plan is to dynamically feed the intrusion and anomaly detection models with knowledge provided by a number of system specific sources, such as topology-, role- and policy-based knowledge, trust-based mechanisms and strategic usage of honeypots.

  Research in these two areas will proceed in the next stages of WP3000, in order to develop and/or choose the best techniques to use in the integrated platform.
- The solution must be scalable in order to be deployed on several CIs while maintaining adequate operational performance.
- It must be minimally intrusive, in such a way that its operation must not interfere with the normal operation of the CI. Moreover, its management and operational overhead in terms of infrastructure resources must be minimal.
- Also, the analysis and detection infrastructure must be designed taking into account local detection mechanisms (able to function autonomously on each component of the industrial control network) and coordinated detection mechanisms, for multi-dimensional distributed intrusion detection and prevention.

5.1.3 Module interaction

Module interaction is processed accordingly with Figure 5-3, which shows an example evolving two CIs. For each CI, there is a Perimeter IDS that receives information from field adaptors and detection agents. There are local detection agents for each field network (in practice, this is a kind of an autonomous system, in such sense that it demarcates an area where autonomous response capabilities, corresponding to smart RTU policies, might be deployed and available).

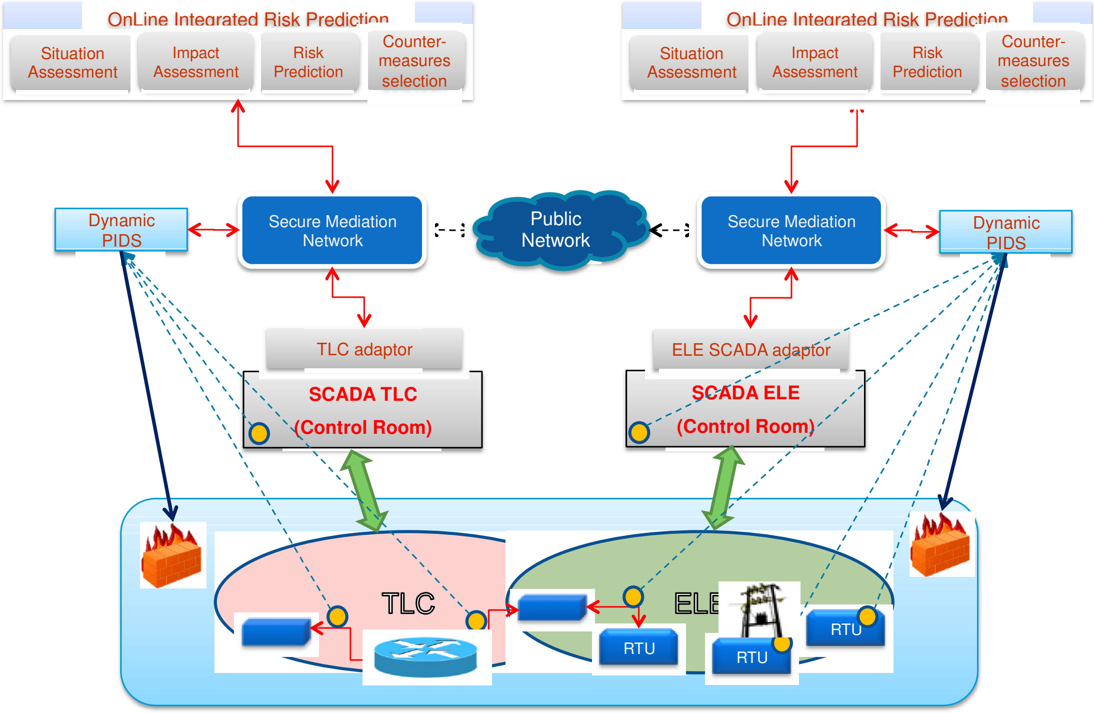


Figure 5-3: CockpitCI architecture component interaction

The PIDS must be able to enforce reaction mechanisms, in response to ongoing threats, for instance through reconfiguration of a firewall (example shown in the figure). However, the present document is concerned with the distributed IDS capabilities of the PIDS and not with its reaction mechanisms.

5.2 Generic probing architecture

The proposed detection architecture (see Figure 5-4) is built on a distributed infrastructure that aggregates several probing and monitoring points, working together on close coordination to provide the surveillance capabilities for the security platform. The functional criterion for deploying those security probes (or sensors) divides the SCADA infrastructure on three different security zones, namely:

- **IT Network** - This is the organization's IT Network. While this network isn't part of the SCADA system, it may host SCADA components, like Human-Machine Interaction

(HMI) consoles. Also, historical evidence has shown that several successful attacks have found its way to the critical SCADA components through this level of the networking infrastructure.

- **Operations/Process Network** - This network hosts the main SCADA components, such as Master Stations, Database Management Systems (DBMS) servers or HMI Consoles.
- **Field/Control Network** - This network hosts the field devices, like Remote Terminal Unit (RTUs), and process sensors.

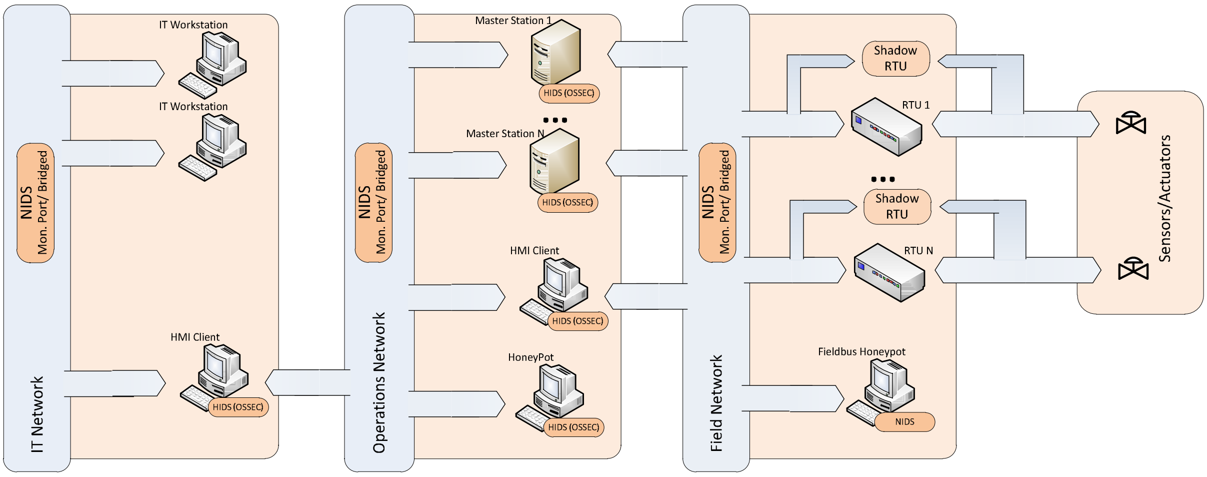


Figure 5-4: Generic probing architecture

This multi-zone topology provides a contextual approach to the problem of probe placement that takes into account the existence of different network scopes, which can be easily segmented and separated by a well-defined perimeter. The criteria for zone separation follow the structure that was presented in section 2.1 of this document.

This separation has two purposes: first, to segment different infrastructure contexts for which different detection, correlation/Inference and reaction strategies might apply; second, to provide well-defined security perimeters between each zone, which are critical to provide mediation mechanisms which may inspect and control information flows between each one.

These security perimeters are strategic positions not only for detection purposes, but also for the reaction and countermeasure mechanisms to be addressed in the next developments of this WP.

5.2.1 NIDS and HIDS techniques

Intrusion Detection Systems (IDSs) provide an additional level of security for networks and systems, by providing critical information about attacks. They can actively block communications or simply monitor a network. In the first case, the so called Intrusion

Prevention Systems (IPSs) have the capability of blocking attacks or preventing exploits from being successful. IDSs fall into two main categories: Host IDS (HIDS), Network IDS (NIDS). The first one is responsible for monitor operations within a single host like file system changes. Whereas the NIDS are responsible for monitor a network segment collecting and analyzing packets in order to detect unwanted communications. Hybrid IDS share the characteristics of NIDS and HIDS. For detection layer in the CockpitCI project both HIDS and NIDS will be used. Since the detection layer conceptually separated from the reaction mechanisms both types of IDS will only monitor and forward the events to the correlation layer. A broad overview of their role is presented in the following paragraphs and further detailed in the deliverables D3.2 [CockpitCI2013], D3.3 [CockpitCI2013a] and D3.4 [CockpitCI2013b].

HIDS focus on intrusion detection on the host-level. This category includes several types of sensors:

1. Log monitors, which parse and process system logs, searching for patterns of suspect activity. Platforms such as the Prelude IDS [Vandoorselaere2008] or OSSEC [OSSEC] include specialized log parsers which can be extended and customized.

2. Integrity monitors that watch key system structures and components for changes, such as registry keys in windows systems or critical files. Tripwire [Spafford1994] and OSSEC [OSSEC] are able to perform these tasks, being able to monitor any change on a system. However, a known safe baseline (starting with clean systems) must be previously established before deploying such solutions, at the risk of considerately reducing their effectiveness.

3. Signature-based sensors have a set of built-in event signatures that can be matched against network traffic and log entries. Mostly reactive by nature, these sensors are also useful to track unauthorized users on hosts.

4. Application behaviour and system call analysers, have the ability to intercept and analyse calls between applications and the operating system in order to detect improper application and system behaviour.

NIDSs, which may include both signature and anomaly-based systems, (next discussed) focus on network-level intrusion detection.

5.2.1.1 IDS Paradigms, the CIDF model and SCADA architectures

The Defense Advanced Research Projects Agency (DARPA) Common Intrusion Detection Framework Architecture (CIDF) [CIDF, Staniford-Chen1998, Tung2001, Kahn1998] was an effort to develop standard protocols and APIs, allowing intrusion detection systems to share information and resources. Many of the ideas developed within the CIDF effort were also the basis for IETFs Intrusion Detection Working Group (IDWG) work, such as the Intrusion Detection Message Exchange Format (IDMEF [Debar2007]), used for interchange of security events.

One of the most noteworthy results of the CIDF effort consisted on the definition of a generic IDS architecture, as shown in Figure 5-5.

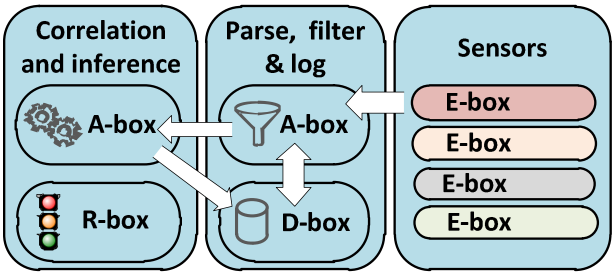


Figure 5-5: CIDF generic IDS architecture

The CIDF IDS architecture builds on discrete functional blocks with clearly defined functions:

1. E-blocks (event-boxes): generic sensor elements that acquire information to be processed by other functional blocks. Network traffic probes, for instance, are an example of such elements.

2. D-blocks (database-blocks): generic data persistence elements which store and persist information from E-blocks, for subsequent processing. Without these elements, IDS architectures would be limited to a simple real-time reactive operation.

3. A-blocks (analysis-boxes): generic processing modules which analyse, correlate and infer information from D, E and even other A-blocks to detect anomalies or suspicious behaviour, being able to generate alarms.

4. R-blocks (reactive-blocks): generic action enforcement blocks, which implement specific actions and countermeasures to deter or avoid a threat. An R-box might be fed by D and A-boxes.

The CIDF model is of particular interest because it offers a generic decomposition tool to analyze the modules a generic IDS architecture.

A-boxes frequently contain sophisticated analysis, correlation or inference mechanisms which commonly distinguish IDS paradigms from each other, whose implementation is the subject of intensive study in the last years. Existing methodologies are usually classified in two main groups [Garcia-Teodoro2009, Douglieris2004]:

1. Signature/fingerprint-based detection is based on characteristics extracted from traffic flows, such as statistical variations of specific parameters (frequently related to traffic volume) or patterns such as the distribution of involved IP addresses or ports. These methods are unsuccessful in identifying unknown anomalies, requiring supervised analysis and/or training to incorporate new signatures in the IDS – this has the side effect of letting the network unprotected from rogue threats for a variable amount of time. Tools such as Snort [Snort] fall into this category when used in its

simplest configuration (without plugins as the Statistical Packet Anomaly Detection Engine (SPADE) [Staniford2002] or OSSEC [OSSEC]).

2. Anomaly-based detection consists on finding deviant behaviour from established “normal” usage patterns. Several techniques have been researched on this field, based on statistical, knowledge-based or machine-learning techniques, using IP-flows, single-link or network-wide data with signal-processing techniques (such as wavelets) [Barford2002, Brutlag2000], Kalman filters [Soule2005], PCA (Principal Component Analysis) [Lakhina2004] or Sketches [Krishnamurthy2003, Dewaele2007]. However, there are two fundamentally different approaches to anomaly detection which distinguish one from another in what respects to their autonomy.

   Anomaly detection based on supervised learning requires training based on labelled traffic, which is normally inconvenient to produce. This helps establishing a baseline model which corresponds to “normal” traffic – any deviating pattern is considered anomalous (in practice this corresponds to behavioural profiling). This method is able to detect unknown anomalies and rogue threats – however the training process is time-consuming and requires a regular feed of anomaly-free data sets (a complex and error-prone task) which must be kept up to date to be effective. The Unsupervised Root. Cause Analysis (URCA) tool [Silveira2010], for instance, uses both signature-based and supervised learning techniques. Another example is presented in [Perdisci2010].

   Autonomous/unsupervised anomaly detection is a somewhat recent trend, based on the assumption that an IDS should not rely on previous knowledge to operate, rather being able to autonomously detect and characterize threats. While some authors [Mazel2011, Mazel2011a] propose that modern networks should rely on completely unsupervised detection and reaction methods, common sense dictates otherwise as a failure could rend inoperable significant sections of the network infrastructure (due to automatic misjudgement and consequent decision). Botminer [Gu2008] is an example of a tool that uses these methods, performing cross-cluster correlation to identify hosts with similar suspicious activity patterns.

A-box choice and positioning criteria must obey some restrictions, especially when used at the network appliance-level. Frequently, network appliances are embedded systems platforms with reasonable but limited computing resources. For instance, among anomaly-detection methods, those based on real-time IP flow analysis using time-slots are found to be particularly adaptable and flexible enough for integration on router-embedded A-boxes.

For unsupervised detection schemes, the majority of published work on the subject is based on sub-space and inter-space clustering anomaly detection methods, for instance using different flow levels for time series analysis (as proposed by [Mazel2011]). Subsequent correlation of anomalies from multiple sources might be performed at a higher level, enabling the possibility of network-wide meta-correlation. As an example, [Mazel2011] proposes performing anomaly correlation from single-link multiple flow aggregations to estimate their impact by finding if it they are visible at different flow levels – this idea could be further extended to network-wide scope if performed at a higher level (as suggested by [Lakhina2004a]). This concept might also be applied for anomaly characterization and autonomous reaction techniques, in which case R-boxes must also be able to generate the adequate action for an autonomously generated threat response.

Other techniques, which are of particular use in HIDS systems, such as target monitoring (used by tools such as Tripwire [Vandoorselaere2008], which control and report changes on internal system files and parameters) can also be supported using OSSEC (although they are not covered in this discussion). Several authors classify some hybrid approaches as new IDS categories, such as the case of stealth probes [Marinova-Boncheva2007], which consist of global correlation and inference procedures carried along prolonged periods of time (months) to detect attacks prepared and executed over an extended time.

Individually, each IDS category has its particular set of benefits and drawbacks, which can be overcome with a combination of different techniques for correlation of data obtained from signature-based and anomaly-based detection mechanisms.

5.2.1.2 Domain-specific IDS

Signature-based NIDS (the most common type) are mostly effective to detect attack patterns such as network scans or malformed packets. However, the lack of AAA (Authentication, Authorization, and Accounting) mechanisms in SCADA systems enables an intruder to easily perform an attack by simply forging network streams which are sent to target devices on the control network [Verba2008]. Therefore, the NIDS must have some sort of context-specific information to deal with SCADA systems.

However, typical SCADA networks have specific characteristics that can be used to provide the IDS with a more complete knowledge of the environment it is working on [Verba2008]. Relatively static topologies and control flows enable the use of mapping the possible connections between different equipment, in terms of protocols, ports and direction of the communication flows. Figure 5-6 shows an example of this approach, where a compromised HMI tries to communicate directly with a slave on the control network (something it's not supposed to do). For such abnormal situations, the IDS could be configured to provide alerts.

Another example has to do with SCADA protocol characteristics. For instance, Modbus frames cannot exceed a maximum size of 256 bytes. As such, it would be relatively easy to an attacker to forge packets to cause a buffer overflow in a slave [Zhu2011]. Since this is possible to achieve while maintaining a correct framing structure for the protocols of the network layer, conventional IDS are not able to detect such attacks. Moreover, if the control protocol frames are correctly forged, an attacker can induce deviant behaviour on the control systems. To overcome these problems, an IDS might be able to assess if a given command makes sense from an inference database with actions and transitions states for the system.

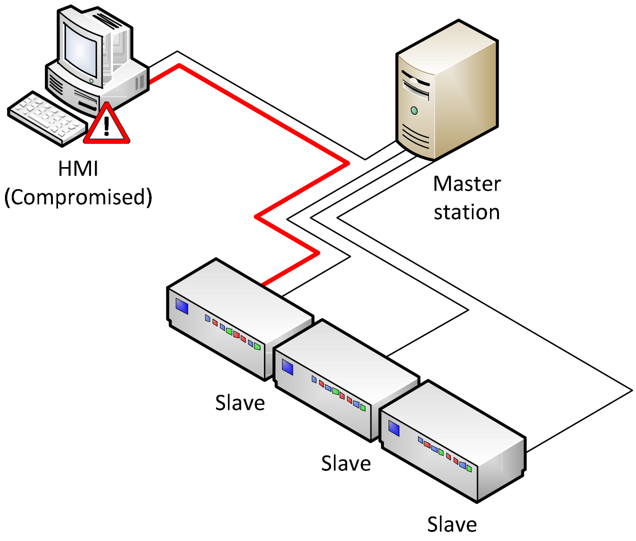


Figure 5-6: Incorrect communication flow in a SCADA system

Figure 5-7 shows an example of a situation where a Master station has become compromised.

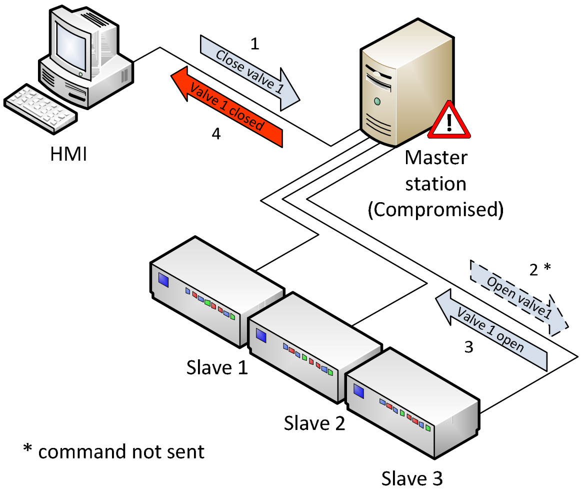


Figure 5-7: Incorrect communication flow in a SCADA system

In this scenario, an operator on the HMI sends a command to close a valve to the master station, which might be modified or not be executed at all, disguising its actions. In those scenarios, an IDS supported by an inference state database could be very effective.

There are SCADA-specific signature packages for the Snort IDS [Snort], as the ones found on [DigitalBond], with support for several mainstream protocols such as DNP3, Modbus and Ethernet/IP. Signature packages for more generic SCADA vulnerabilities are also available.

In this line of thought, another IDS which shows promise is Bro [Bro], a NIDS developed at the Lawrence Berkeley National Laboratory (LBNL), which has the ability to perform disambiguation and analysis of application level semantics, providing context-awareness for IDS detection (which is something of great value when it comes to domain-specific usage) – however, its adaptation for usage in SCADA environments is still a work in progress.

Also, IDS devices, along the electronic perimeter, can establish a baseline profile of the normal system behaviour. In addition, a perspective on an intrusion can be developed by analysing the emerging characteristics of the data such as patterns, clusters and trade-offs by looking for trends and cycles in the data flow. This would require domain specific knowledge of the SCADA network and the associated communication devices in order to construct the IDS attack signature database. Identifying these attack scenarios and generating signatures that correspond to these situations is a significant challenge in itself and would need extensive and detailed analysis of the various attacks in the context of interconnected grids. However, once this is achieved, the observed behaviour needs to be correlated and analysed to detect potential intrusions and filter the attack traffic. The solution of domain specific IDS overlay network, along an extended secure cyber perimeter, which functions in a collaborative manner, has the potential to tackle known cyber attacks to date in a fairly effective manner. It would follow the principle, “Stop the attack even before it reaches you”

5.2.1.3 CockpitCI Network Intrusion Detection Systems (NIDS)

On the edge of each zone described in Figure 5-4, there is a Network Intrusion Detection System (NIDS) that provides the network-level mechanisms for monitoring the data exchanged between adjacent zones or external entities, as well as communications within a network segment. It monitors the network traffic to detect suspicious activity such as probe scans, DoS attacks or MITM attacks. As an example, the operations network NIDS can be used to identify attacks against the control servers and the human-machine interface (HMI), systems that users on the enterprise can access. NIDS can use pattern-based, knowledge-based and anomaly detection techniques to identify and track suspicions network activity.

These NIDS can be implemented in two different ways: either by integrating them within network perimeter firewalls or by making use of networking equipment to place them in monitoring ports that mirror network traffic from other ports (for instance, the port used by the perimeter firewall). The first approach has the benefit of providing close integration between the NIDS and the firewall, which is a device involved in security reaction mechanisms, while the latter enables NIDS deployment with minimal overhead and an almost zero footprint in terms of network visibility. The latest approach will be used providing not only a concept separation between detection and reaction but also a minimal impact on the network flow. In the case of a packet flood or other specific network attack against the system, those detection agents will not affect the availability of SCADA communications since they are deployed in a passive mode. In the same way, network attacks targeting them may disrupt the intrusion events reporting but not SCADA communications, enforcing availability over security.

An open source solution, Snort [Snort], will be deployed as network intrusion system for all network segments. Each NIDS is configured according to its location to perform rule based detection. For instance the NIDS used for the SCADA control network perimeter (eventually a Domain-specific NIDS) will be loaded with adequate rules for protocols used in such environments and moreover, with rules about traffic flows between Masters and Slaves.

Snort behaviour is composed by set of steps better described in the deliverable D3.3. It starts with a packet decoding via libpcap [Libpcap] followed by a pre-processing stage. After these two first steps, the packet already represented in a logic set of field and parameters can be further processed. In the next step, the detection phase, the stream of packets is matched against the conditional set of rules. Finally, and in the case of an intrusion, snort triggers a set of alarms. Those alarms are post processed, converted to the IDMEF format and forwarded to the local correlation engine.

5.2.1.4 CockpitCI Host Intrusion Detection Systems (HIDS)

Host-based intrusion detection systems rely on events collected on the hosts they monitor, as opposed to Network Intrusion that collect their input data by monitoring network traffic.

A typical HIDS uses systems logs as one source of information to detect attacks on specific environment. Log files record the behaviour of computer system and aims at recording the action of operating system, applications, and use behaviours. The logging system is widely used for system debugging, monitoring, and security detection, thus is particularly important in intrusion detection.

Another way of a HIDS to gather information from a host is by monitoring the behaviour of applications by observing the interaction of those applications with the operating system. The interaction between the applications and operating system usually takes place by system calls, this is the way a program requests a service from an operating system.

The HIDS analyses the system calls and identifies threats by comparing the signatures of these calls with the signatures from system calls made by known attacks, or uses machine learning methods to learn the normal behaviour of applications system calls and recognizes possible attacks by looking for anomalies.

A HIDS also check a system for anomalies like the presence of hidden ports, unusual file permissions and inspect all running processes to find an abnormal process running.

There are a multitude of attack types and many attack vectors, but one thing is common to all them, an attacker usually leaves traces and changes the system in some way. From viruses that modify a few files, to kernel-level rootkits that alters the kernel, there is always some change in the integrity of the system. The integrity checking is thus an important part of intrusion detection by detecting the integrity of the system.

Usually, the integrity checking is done by the HIDS by storing the cryptographic hashes of files, configurations or Windows registry entries, in a database. This initial step is made when the system is known to be in a good state. Then a daemon runs periodically to check if any of the hashes of the current files has changed when compared to the hash stored in the database. A changed hash means that the file was modified, as two different files cannot have the same hash.

Host IDS (HIDS – which attempt to detect intrusion at the host level) are to be deployed in the main components of the SCADA system, e.g., Master Stations, DBMS, HMI consoles, in order to track host-level unauthorized or suspicious activity.

Regarding the deployment of HIDS, there are some precautions to be taken especially for devices on the operations network to ensure that the HIDS does not poses an excessive overhead into a SCADA control system. As to a HIDS, there are risks involved with adding software to a control server and with automated response, similar to the problems of antivirus software. Some of the HIDS software inserts itself in the TCP/IP stack and acts as an intermediary, potentially causing a number of performance and timing issues. Vendors and users must certify the HIDS agent will not conflict with the SCADA application.

OSSEC [OSSEC] is the HIDS considered for inclusion in the CockpitCI intrusion detection architecture. This tool is an open source multiplatform HIDS. It can perform log analysis, integrity checking, Windows registry monitoring, rootkit detection and real-time alerting.

5.2.2 Honeypots and honeynets

A Honeypot is a decoy or dummy target set up to attract and detect/observe attacks. By being exposed to probing and attack, its purpose is to lure and track intruders as they advance. Deploying and running a honeypot infrastructure requires a careful approach: defences have to be planned in advance so that the infrastructure itself cannot be used to increase the attack surface, while keeping a low profile.

A Honeypot can be implemented in a different fashion, depending on its operation scope: in the operations network a honeypot might simulate the operation of a network server (e.g., Master Station), while in the field network a honeypot could be implemented using a system capable of simulating the operation of an RTU (e.g., a Modbus emulator).

Honeypots can be classified in two groups: research and production – the first are used to obtain intelligence information about attack methods, while the latter is used to implicitly protect and ICT infrastructure by providing advance warning of attacks against the production infrastructure. Honeypot types can also be distinguished by the ability of the attacker to interact with the application or services [Spitzner2002]:

- High-interaction honeypots can be probed, attacked and compromised. These honeypots let the attacker interact with the system in order to capture the maximum amount of information regarding his intrusion and exploitation techniques. Consequently, these honeypots have no restrictions regarding what the hacker can do, once the system is compromised and, as such, require a lot of close monitoring and detailed analysis.

- Low-interaction honeypots [Provos2004] emulate vulnerabilities rather than presenting real ones, therefore restricting the attacker's ability to interact with it. Mainly used as decoys, they are also less flexible, albeit being more secure since there is little that the attacker can do. Nephentes [Baecher2006] or Honeyd [Honeyd] are examples of this honeypot type.

Still regarding honeypot types, there is also another distinction [Riden2010] that can be established between server and client honeypots. The first type is designed to passively wait for attacks, while the latter is able to actively search for malicious servers and behave like a victim (useful for detecting client-side browser exploits). Examples of client honeypots are the Shelia [Shelia], Honeymonkey[Wang2006] and CaptureHPC [Hes2009].

Honeynets extend the concept of Honeypots in a distributed fashion, by deploying several honeypot instances on a production network. This requires at least two components: a Honeywall and Honeypot hosts. In these situations, an attacker has access to a high-interaction Honeypot (with a full-fledged OS) – however, in order to limit the possibility of an attack, the honeywall (which also maintains an internal IDS to monitor an track suspicious activity) acts as firewall (ideally operating in bridging mode, without having an IP on the network, apart from the management interface), limiting outbound connections or even using a "bait-and-switch" technique to reroute traffic to another host.

The Honeynet project [Honeynet] defines two architectures: Gen I and Gen II. The first one, which is nor able to conceal its existence, proved to be vulnerable to discovery and probing by skilled attackers, being easy to fingerprint – also, there are no sensors on the Honeypot operating systems.

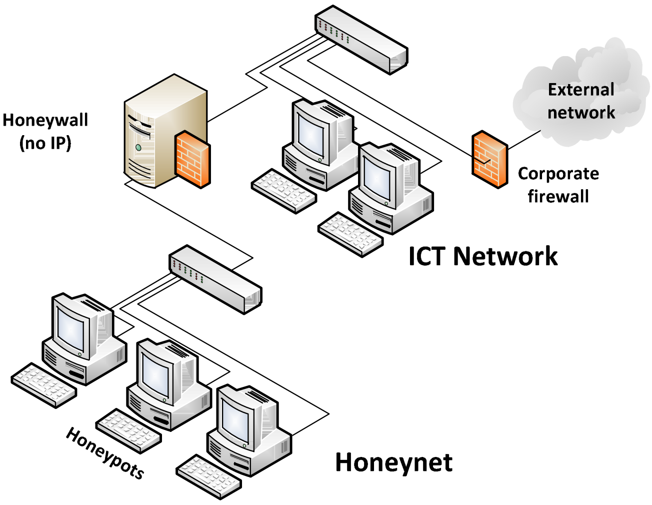


Figure 5-8: Gen II honeynet topology

Gen II honeynets (Figure 5-8) are harder to detect, being designed with stealth capabilities. Honeypots include recording on the host side, even on encrypted connections, also incorporating keylogging capabilities. Honeywalls are implemented as Layer-2 firewalls, which are harder to detect and fingerprint since they act as transparent bridges, connecting the Honeynet to the production networks, maintaining the same address range.

5.2.2.1 Security categories

To assess the value of Honeypots we will break down security into three categories as defined by Bruce Schneier in Secrets and Lies [Schneier 00]. Schneier breaks security into prevention, detection and response.

Prevention

Prevention means keeping the bad guys out. Normally this is accomplished by firewalls and well patched systems. The value Honeypots can add to this category is small. If a random attack is performed, Honeypots can detect that attack, but not prevent it as the targets are not predictable. One case where Honeypots help with prevention is when an attacker is directly hacking into a server. In this case a Honeypot would cause the hacker to waste time on a non-sufficient target and help preventing an attack on a production system. But this means that the attacker has attacked the Honeypot before attacking a real server and not otherwise. Also if an institution publishes the information that they use a Honeypot it might deter attackers from hacking. But this is more in the fields of psychology and quite too abstract to add proper value to security.

Detection

Detecting intrusions in IT networks is similar to the function of an alarm system for protecting facilities. Someone breaks into a house and an alarm goes off. In the realm of computers this is accomplished by Intrusion Detection Systems or by programs designed to watch system logs that trigger when unauthorized activity appears.

The problems with these systems are false alarms and non-detected alarms. A system might alert on suspicious or malicious activity, even if the data was valid production traffic. Due to the high network traffic on most IT networks it is extremely difficult to process every data, so the chances for false alarms increase with the amount of data processed. High traffic also leads to non-detected attacks. When the system is not able to process all data, it has to drop certain packets, which leaves those un-scanned. An attacker could benefit of such high loads on network traffic.

Response

After successfully detecting an attack we need information to prevent further threats of the same type. Or in case an institution has established a security policy and one of the employees violated against them, the administration needs proper evidence. Honeypots provide exact evidence of malicious activities. As they are not part of production systems any packet sent to them is suspicious and recorded for analysis. The difference to a production server is that there is no traffic with regular data such as traffic to and from a web server. This reduces the amount of data recorded dramatically and makes evaluation much easier. With that specific information it is fairly easy to start effective countermeasures.

5.2.2.2 Honeypots in the CockpitCI cyber-analysis and detection layer

The architecture includes at least one honeypot for each network zone, which is able to simulate the behaviour of a certain component, in order to detect illegal interactions. In the case of a honeypot, all interactions are illegal, since it does not perform any real function within the infrastructure, serving as bait, operating in the unused network address space.

A honeypot can be implemented in a different fashion, depending on its operation scope: in the operations network a low-interaction honeypot might simulate the operation of a network server (e.g., Master Station), while in the field network a honeypot could be implemented using a system capable of simulating the operation of an RTU (e.g., a Modbus emulator). In the process or ICT network, high-interaction honeypots might be adequate (even in the form of virtual machines, co-located on a same host), as well as low-interaction honeypots simulating minimal services.

Also, some attacks targeting the system can be redirected to the honeypot, therefore providing more information about the attacker and his intentions

5.2.2.3 IT network honeypot

The concept of the IT network honeypots is to catch malicious IT network activity with a prepared machine. This computer is used as bait. The intruder is intended to detect the Honeypot and try to break into it. Next the type and purpose of the Honeypot specifies what the attacker will be able to perform. Often Honeypots are used in conjunction with Intrusion Detection Systems. In these cases Honeypots serve as Production Honeypots and only extend the IDS.

A common setup is to deploy a Honeypot within a production system. **Production Honeypots** are primarily used for detection (see 5.2.2.1). Typically they work as extension to Intrusion Detection Systems performing an advanced detection function. They also prove if existing security functions are adequate, i.e. if a Honeypot is probed or attacked the attacker must have found a way to the Honeypot. This could be a known way, which is hard to lock, or even an unknown hole. However measures should be taken to avoid a real attack. With the knowledge of the attack on the Honeypot it is easier to determine and close security holes. A Honeypot allows justifying the investment of a firewall. Without any evidence that there were attacks, someone from the management could assume that there are no attacks on the IT network. Therefore that person could suggest stopping investing in security as there are no threats. With a Honeypot there is recorded evidence of attacks. The system can provide information for statistics of monthly happened attacks. Attacks performed by employees are even more critical. Typically an employee is assigned an IT network account with several user privileges. In many cases networks are closed to the outside but opened to the local network. Therefore a person with legal access to the internal IT network can pose an unidentifiable threat. Activities on Honeypots can be used to proof if that person has malicious intentions. For instance an IT network folder with faked sensitive documents could be prepared. An employee with no bad intentions would not copy the files but in the case the files are retrieved this might reveal him as a mole.

Another benefit and the most important one is that a Honeypot detects attacks which are not caught by other security systems. Section 5.2.2.1 gives a more detailed description on how a Honeypot can help detecting attacks.

Research Honeypots are used in a different scenario. A research Honeypot is used to learn about the tactics and techniques of the Blackhat community (in the computer security community, a Blackhat is a skilled hacker who uses his or her ability to pursue his interest illegally). It is used as a watch post to see how an attacker is working when compromising a system. In this case the intruder is allowed to stay and reveal his secrets.

The Honeypot operator gains knowledge about the Blackhats tools and tactics. When a system was compromised the administrators usually find the tools used by the attacker but there is no information about how they were used. A Honeypot gives a real-live insight on how the attack happened.

5.2.2.4 Operations network honeypot

A honeypot is planned for the Operations Network. This Honeypot follows the same design as the IT network honeypot. Its purpose is to act as a decoy to the attackers, detecting their presence in the system. For this honeypot, traditional software such as Honeyd [Honeyd] can be implemented. It will simulate services commonly found in ICT networks, e.g., File Transfer Protocol (FTP) or HTTP services, allowing an attacker to interact with it. Interaction from attackers will generate alerts. These security events need to be transmitted to the local correlator for further analysis. To achieve this goal, a security adapter needs to be developed for processing and transmission of those events. The security adapter role in this honeypot is to read the output of the honeypot, parse the information to the correct format used by the detection layer architecture, and send it to the local correlator. This security adapter will be able to connect with the security management platform for ease of configuration and management operations.

5.2.2.5 CockpitCI field network honeypot

The field network honeypot needs a different approach in comparison with the previously presented IT and Operations Networks honeypots in sections 5.2.2.3 and 5.2.2.4. Mainly, the natures of the networks differ from each other. While the previously ones are an ICT environment, the field network hosts, mainly, the field devices (PLCs) of the ICS/SCADA system. This calls for a honeypot capable of detecting attacks and threats to this specific area. With that in consideration, it is proposed a honeypot capable of simulating the behaviour of a control device (PLC). To achieve this goal the field network honeypot will simulate the behaviour of the Modbus TCP protocol [Modbus]. If an attacker tries to exploit it, it will trick him/her into thinking it is interaction with a normal Modbus TCP enabled PLC and will log the interaction. Moreover the honeypot can simulate other services commonly available in production honeypots, for example: FTP and Simple Network Management Protocol (SNMP) services. An additional port scan service (not directly related to ICS environment security) will be added to detect interactions with application ports other than those related with PLC simulation. This allows the honeypot to have a broader range of detection. The security events raised by the honeypot will be transmitted to the remaining detection architecture, namely, its local correlator.

To accomplish this objective, the Modbus will follow a modular design where each module performs a given task. It will have a module for each service simulated, and additional modules to perform tasks regarding event processing such as event filtering and transmission. For ease of configuring and managing the honeypot, a management module is included. This module will aid the task of performing tasks such as configuring the detection and event processing modules. It will have an interface to interact with the security management platform of the detection layer architecture. A modular architecture has the advantage of allowing additional modules to be easily added to the field network honeypot in the future.

As in the honeypot scenarios presented in the previous sections, the honeypot range of action must be contained [Li2011]. Failing to do so may result in a scenario where the attacker can gain access to the ICS/SCADA system allowing him/her to wreak havoc. In order to prevent such scenario, a firewall must be placed between the honeypot and the remaining network. A Layer-2 firewall like the one presented in section 5.2.2 (Figure 5-8) can limit the interaction of the honeypot to the attacker while protecting the system and remain undetected. An example scenario is illustrated in Figure 5-9.

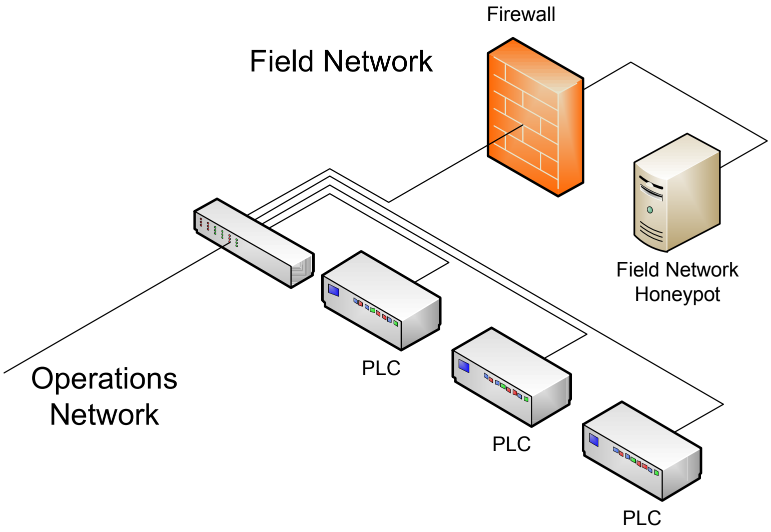


Figure 5-9: Field network honeypot placement

5.2.3 Shadow RTUs

In order to monitor the interaction of an RTU with its field network, a shadow RTU can listen to its communications (see Figure 5-10). This task can be accomplished by a device with somewhat limited processing and I/O capabilities, for instance, a device based on a microcontroller (like an Arduino), or a single-board computer (like the Raspberry PI) [RaspberryPi]).

The shadow RTU concept has its roots in the aeronautic industry, were multiple redundant control and monitoring systems are the norm. It is frequent to find several duplicate systems in modern airplanes just for the sake of reliability and continuous, fail-safe operation.

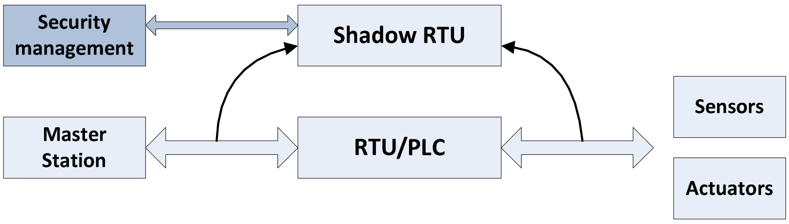


Figure 5-10: Shadow RTU interaction

In Figure 5-10, the shadow RTU has access to the inputs and output states of an RTU (this can be accomplished by simple Modbus register queries, or by placing the shadow RTU in a monitor port of a switch to intercept incoming commands - eventually, interception of output signals using optical decoupling or other techniques may be feasible. Another option is interception with a passive Ethernet tap device which is an even more efficient way to capture all packets flowing across the system). The Shadow RTU is also connected to the security management infrastructure by means of a separate network, used to report security status information.

The Shadow RTU concept also goes in line with the Smart RTU layer concept, opening the way for supporting complementary sophisticated mechanisms for resilience and self-healing.

For simple, discrete I/O on Ethernet control networks using Modbus/TCP, a Shadow RTU can be easily built using a compact, low-cost, Single Board Computer with an Ethernet interface and Linux OS. By using open-source libraries (such as pvbrowser [Pvbrowser]) it is possible to poll in parallel the status of the monitored RTU – something that is possible in Modbus/TCP since it operates on a peer-to-peer model.

5.3 The CockpitCI cyber detection and analysis architecture

This chapter describes the proposed detection architecture for the CockpitCI (Figure 5-11), which builds on the generic probing architecture from the previous chapter to provide a complete security detection solution. Particular relevance is given to the distributed multi-zone, multi-level correlation structure that processes the information provided by the security sensors, complemented by machine-learning capabilities. Once again, it is important to remember that this architecture does not contemplate any kind of specific reaction mechanisms, which are to be addressed in next developments of WP 3000.

As specified in the requirements (and hinted by the generic probing architecture), the proposed architecture for the cyber-analysis and detection layer makes use of sensors for network traffic, host and field device and network activity monitoring, also including analysis components (in the form of correlators and machine learning mechanisms) that implement complex detection patterns, to search for anomalies and abnormal activity. The proposed architecture also provides mechanisms for execution of reaction countermeasures; in the case a security event requires a fast reaction, within a limited time window.

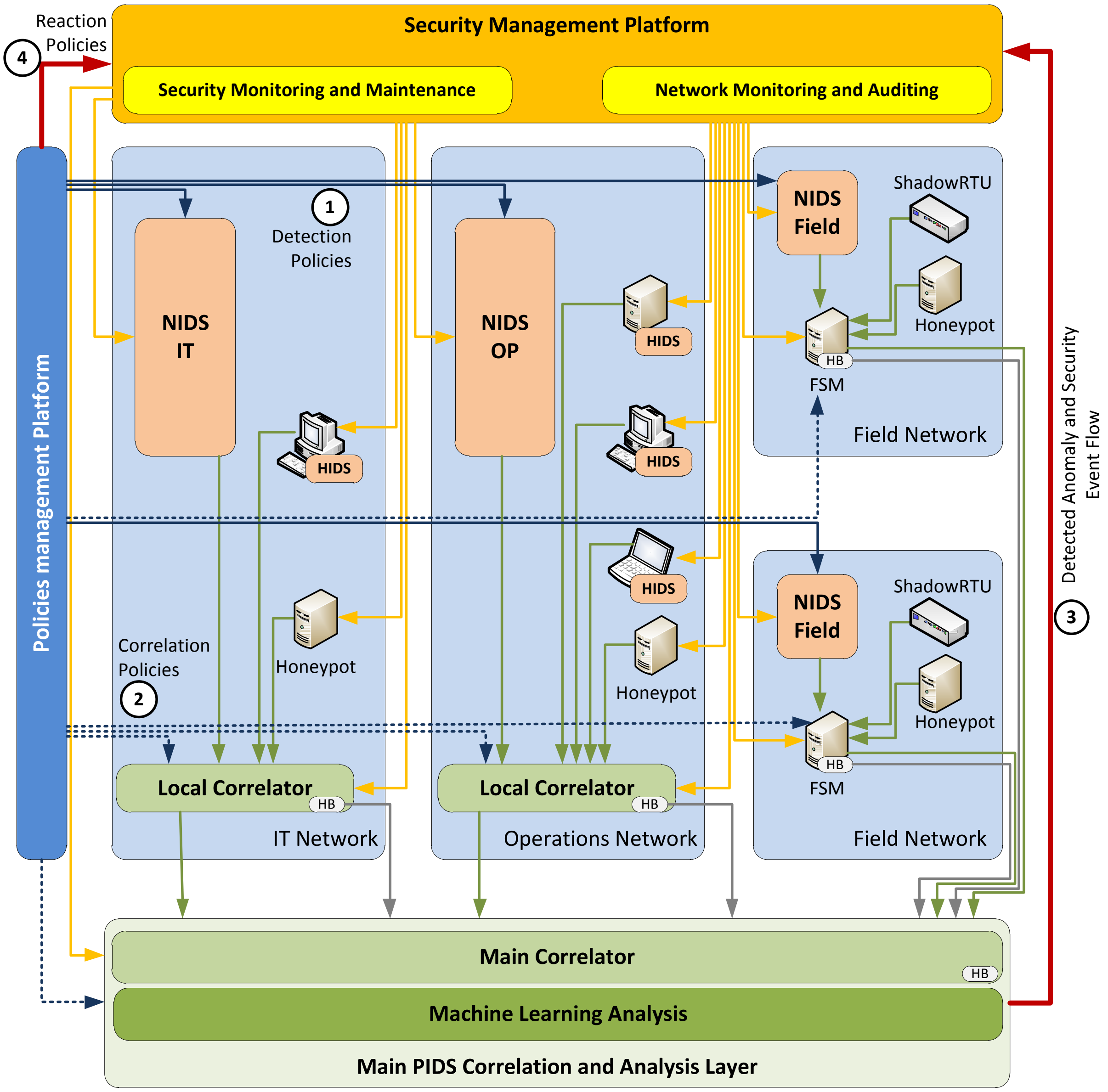


Figure 5-11: Proposed CockpitCI detection and analysis architecture
(Security management flows in red, network security information flows in green)

This architecture was designed with several attack scenarios in mind, namely:

- Sending of unauthorized commands to control equipment;
- Sending of false misleading information to operators, to intentionally lead them to make wrong decisions;
- Perturbation of the system operation, delaying or blocking control flows in the control network;
- Non-authorized modification of control equipment, through manipulation of alarm thresholds and other configurations;
- Interference with available resources due to propagation of malicious software (e.g. virus, worms).

This architecture also presents some innovative concepts for capture and analysis of security information, namely the Shadow RTU and BMS (Backup Master Stations), which will be next described.

Due to the demanding availability requisites and little tolerance to delays, the detection architecture is to be implemented using a network that is separate from the SCADA system network (eventually it can use the same physical network, using VLAN (Virtual Local Area Network) or other types of overlay techniques for traffic separation), in order to guarantee that it does not interfere with the normal operation of the control network.

5.3.1 Event correlation for security detection

Event correlation is a procedure where a stream of events is processed, in order to detect (and act on) certain event groups that occur within predefined time windows. Particularly, Security Event Correlation is of particular interest in the context of the CockpitCI project, being used in the reference architecture described in Chapter 5.

Event correlation, from a security standpoint is complementary to existing security countermeasures, especially for incident detection, analysis and response. These security events must be collected and analyzed from as many sources as possible in order to assess threat and formulate appropriate response. In fact, as pointed out by SANS [SANS], deploying and analyzing a single device in an effort to maintain situational awareness with respect to the state of security within an organization is very limitative approach.

Event correlation usually takes place inside one or several management platforms (also known as Network Management Stations or Network Management Systems). It is implemented by a piece of software known as the event correlator. This tool is fed with events originating from managed elements (system logs, for instance) or monitoring tools.

Each event captures information in a domain of interest to the event correlator (e.g., an anomalous traffic flow, unrelated network accesses, a traffic surge, just to mention a few examples). The event correlator plays a key role in the integration of management, for only there do network, system and service events come together. For instance, this is where the failure of a service can be ascribed to a specific failure in the underlying IT infrastructure.

An event may convey an alarm or report an incident (which explains why event correlation used to be called alarm correlation), but also if situation is transient and goes back to normal, or simply send some information that it deems relevant (e.g., policy P has been updated on device D). The severity of the event is an indication given by the event source to the event destination of the priority that this event should be given while being processed.

Event correlation can be decomposed into four steps: event filtering, event aggregation, event masking and root cause analysis.

- **Event filtering** discards events that are deemed to be irrelevant to the event correlator, such as informational messages (e.g., printer X needs paper in tray).
- **Event aggregation** (also known as event de-duplication) attempts to reduce the number of events, by merging duplicates of the same event (sometimes caused by repeated reporting of the same issue).
- **Event masking** (also known as topological masking in network management) consists in ignoring events related to entities that are behind of a failed system. For example, servers that are downstream of a crashed router will fail availability polling.
- **Root cause analysis** is the last step of event correlation, consisting in analyzing dependencies between events, based on a model of the environment and dependency graphs, to detect whether there is a cause-effect relation between them.

At the end of these stages the correlation process is finished, from a formal standpoint However, some event correlators found on the market sometimes also include problem-solving capabilities, being able to automatically initiate corrective actions.

To better understand how correlation mechanisms operate, this subsection will analyse an example [SANS], illustrated by Figure 5-12.

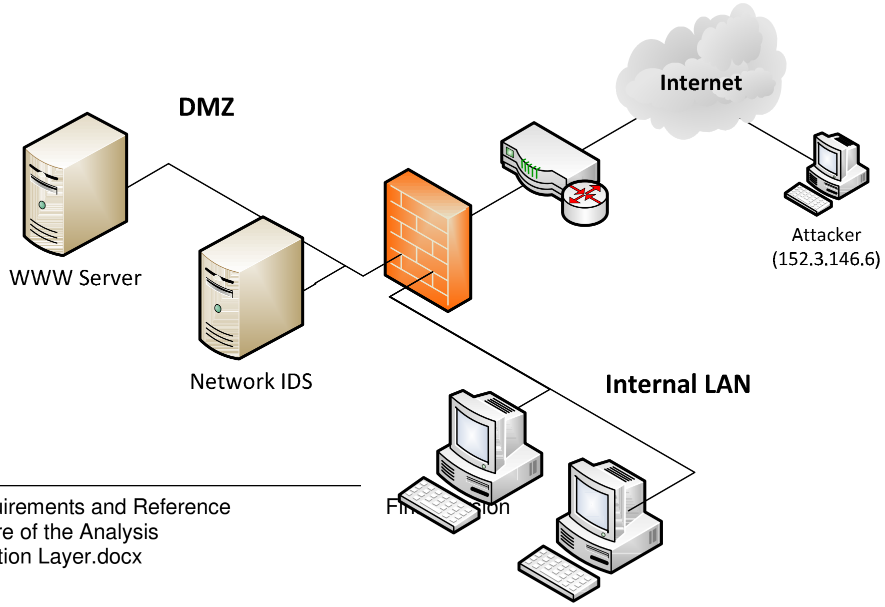


Figure 5-12: Example network scenario for correlation analysis (Adapted from [SANS])

This scenario shows how independently obtained security events can be correlated from several event sources, providing a higher-level perspective for intrusion analysis. The example network follows a commonly found network layout where countermeasures such as firewalls and intrusion detection systems are deployed.

In this example there are several log sources:

- **Router**: Access control lists (ACL's) provide perimeter packet filtering with (typically) syslog-style alerting. They are a first line of defence since they implement traffic policies. When ACL's are combined with logging, it becomes possible to detect reconnaissance and probing activities.
- **Firewall**: modern firewalls can have extensive logging capabilities. For instance, application-level firewalls (like HTTP proxies) can provide both access control and network activity log capabilities, offering a broad view of the network perimeter.
- **Network IDS**: a NIDS monitors network traffic, looking for suspicious activity that can be logged and reported. By comparing activity patterns with the information from vulnerability databases, NIDS can find and report suspicious activity. However, by nature NIDS alerts tend to include a considerable amount of false positive events – is such situations correlation is very useful to filter relevant issues.
- **Application servers (e.g. www, ftp, email)**: for these devices, server and service activity logs are very valuable, especially considering the fact that they are a preferred target for malicious activity.

For this discussion, we will assume the devices have been configured with full logging capabilities such that maximum visibility is attained. For example, the firewall is configured to log both accepted and denied attempts.

In this scenario, Attacker1 (at 152.63.146.6) is launching a series of probes looking for exploitable Common Gateway Interface (CGI) scripts. The Web Server is an Apache web server running on a typical Linux distribution. For this exercise, we will confine the probes to three well known exploits:

- Common Vulnerabilities and Exposures (CVE)-1999-0067: CGI phf program allows remote command execution through shell metacharacters.
- CVE-1999-0172: FormMail CGI program allows remote execution of commands.
- CVE-1999-0936: BNBSurvey survey.cgi program allows remote attackers to execute commands via shell metacharacters.

Now, we must consider a sequence of events that supposedly were detected on different devices. For this purpose, two separate episodes of activity are involved in the security incident perpetrated by the attacker:

1. On May 31st, host 152.63.146.6 conducted a broad scan of the xxx.yyy.zzz.0/24 network likely in search of web servers (confirmed by router). The interior devices

would not have seen this activity because of strong access control lists on the router. The xxx.yyy.zzz.0/24 network is the only public IP address space assigned to the company so it uncertain if this scanning activity is targeted at this company.

2. On June 1st, host 152.63.146.6 attempted three distinct, and only three, http access attempts against the company web server xxx.yyy.zzz.4 (confirmed by the firewall and web server access_log).
    a. These connection attempts requested the phf, formmail, and survey.cgi CGI scripts (confirmed by firewall, web server access_log, and network IDS).
    b. These connection attempts failed (confirmed by web server error_log). It is therefore unlikely that a system compromise has occurred on the company web server.

The Venn Diagram from Figure 5-13 shows how events reported by individual network devices contributed to form a complete situational awareness perspective through correlative analysis. It also shows that the removal of single source of log data has a decisive impact on the analysis of the security incident.

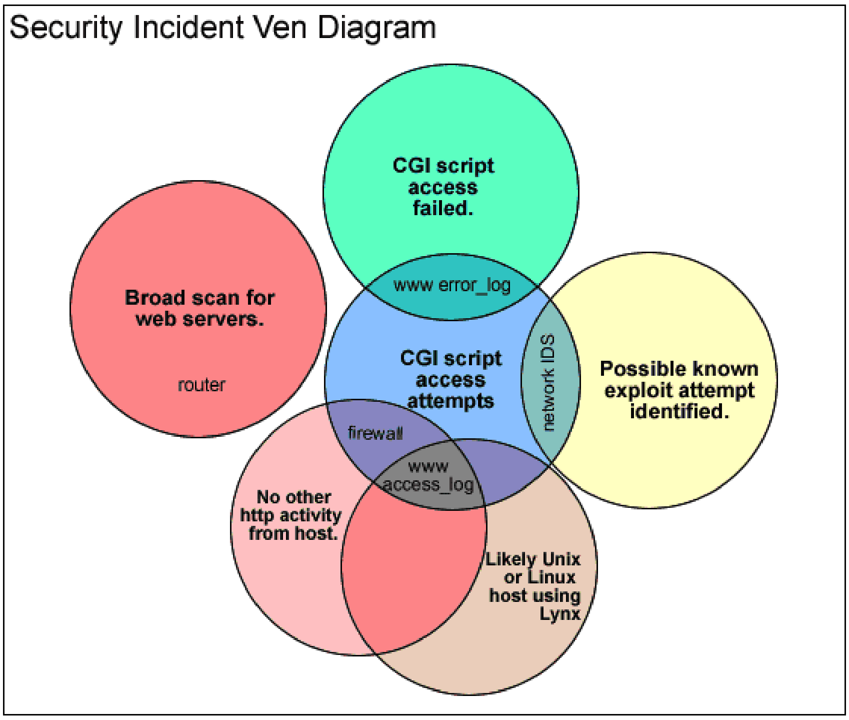

Security Incident Ven Diagram
CGI script access failed.
Broad scan for web servers.
router
www error_log
CGI script access attempts
network IDS
Possible known exploit attempt identified.
firewall
www access_log
No other http activity from host.
Likely Unix or Linux host using Lynx

Figure 5-13: Venn diagram for the correlation process (from [SANS])

Correlation analysis must involve as many sources as possible in order to assess threat and formulate appropriate response. Extraordinary levels of security awareness can be attained in an organization's network by simply listening to what its devices are reporting..

5.3.1.1 Distributed IDS and distributed security event correlation

The concept of distributed IDS with cooperative functions spread among distinct locations is not new: [Cuppens2001] already proposed a Distributed Intrusion Detection System (DIDS) using IDMEF as a means to allow event reduction and interchange. [Antoniadis2002] also presented an IDS based on a distributed probe architecture, with centralized event processing and correlation capabilities. [Ioannidis2002] proposed a router-based Distributed Denial of Service (DDoS) defense system built with cooperating routers which communicate using a special pushback protocol – attacks are traced step-by- step closer to their sources and their bandwidth allocation controlled. [Koutepas2004] described an IDS/IPS framework for DDoS attacks built around a distributed management architecture and based on communities of peers, using multicast to exchange IDMEF messages – this approach assumes that each peer (designated as a Cooperative DDoS Entity) corresponds to ISP. Attack alerts are communicated within the Distributed IDS using a flooding mechanism – once attack detection has been established they install rate limiting filters to fight. [Wan2002] also proposed a DDoS defense solution using strategically placed probes that communicate events through IDMEF messages and are able to activate traffic-limiting mechanisms in order to deter an incoming threat. [Alfaro2006] proposed a DIDS based on a publisher/subscriber model that uses IDMEF for event information exchange, decoupling the publisher, consumer/subscriber and broker/router functions in order to enhance its scalability.

Concerning the optimization of probe distribution and location in distributed IDS, an interesting study was presented by [Suh2005].

Even if several of the approaches that were previously listed explore the idea of spreading traffic probes along the network infrastructure, most of them do not solve the problem of processing and correlating all the collected data in real-time, which constitutes a significant bottleneck in such scenarios. This issue has been specifically addressed in the scope of DIDS distributed inference mechanism research, which was originally born from the need to both detect coordinated attacks at a global scale [Katti2005] and enhance the scalability of those systems, being classified in two different categories [Feamster2010]:

- Anomaly-based detection techniques basically attempt to improve centralized traffic anomaly detection systems by using a central coordinator point that performs large scale inference [Lakhina2004].
- The use of correlation in autonomous detection and decision scenarios was also proposed by [Cuppens2002], who also incorporates historic mechanisms and the capability of analyzing incomplete event chains, through virtual alerts.

Data-sharing techniques use collaborative information sharing [Allman2006, Allman2008] related to several aspects, from network-level indicators [Bailey2005, Cooke2005] to message contents (for mail systems) [Damiani2004, Kong2006, Razor, Pyzor, DCC]. In the scope of these solutions, aggregation and data reduction techniques were also researched, in order to enhance scalability.

In the CockpitCI project, the correlation layer is composed by a local correlator in each network segment and a global scope (main) correlator. The last one uses the output of local correlators in order to get more comprehensive information of the network. Local correlators will be implemented using SEC [Vaarandi2013] or alternatively NodeBrain tool [Boeing2013]. For the global correlator engine will be used the NodeBrain tool. An overview of those tools is provided in the next topics followed by a detailed description in the deliverable D3.2 [CockpitCI2013].

As remark, it should be mentioned that the provided list of methods and techniques is by no means exhaustive. Deliverable D3.2 [CockpitCI2013] provides a more exhaustive description of the correlation techniques and their role in the CockpitCI project. Namely the rule based approach used by local correlators (for the first level of event reduction and processing) as well as the complex correlation and anomaly (machine-learning) based mechanisms used in the main PIDS correlation and analysis layer.

5.3.1.2 Event correlation tools

This section deals with a set of event correlation tools that have the potential for be used in the CockpitCI cyber-analysis and detection architecture.

The **Prelude IDS** platform [Vandoorselaere2008] was designed to deal with event management and correlation. Prelude is a hybrid IDS platform that combines host-based (HIDS) and network-based (NIDS) capabilities, while also including sophisticated programmable event processing and correlation mechanisms [Chifflier2008]. It is extensible and includes a library with Application Programming Interfaces (API) for various programming languages (*libprelude*). In the Prelude platform all security events are encoded using IDMEF, a standard format that can, for instance, be easily forwarded to SQL databases.

Figure 5-14 illustrates the Prelude IDS platform architecture. The central entity for event processing is the Prelude Manager, which accepts data from sensors managed by the *libprelude* library. Sensors are small agents capable of collecting information from several sources, like the Prelude LML (*Log Management Lackey*) that generates events from processing system logs (generating IDMEF events in the form of IDMEF messages that are sent to the event management module through secure SSL connections). Integration with the OSSEC HIDS is also possible, thanks to a plugin agent that directly generates IDMEF events, enabling cross-platform event gathering (which could be used to gather information from Windows-based desktops in the home network, for instance).

The event correlation engine (Prelude Correlator) is able to process IDMEF messages and generate alerts (also in the format of IDMEF messages), using a set of rules described using the LUA or Python programming languages.

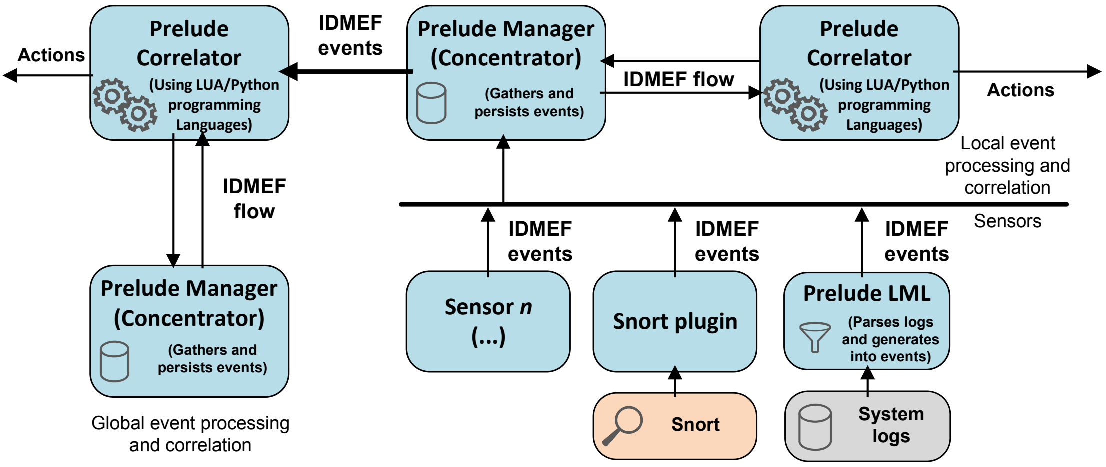

Actions
Prelude Correlator
(Using LUA/Python programming Languages)
IDMEF events
Prelude Manager (Concentrator)
(Gathers and persists events)
IDMEF flow
Prelude Correlator
(Using LUA/Python programming Languages)
Actions
Local event processing and correlation
Sensors
IDMEF flow
IDMEF events
IDMEF events
IDMEF events
Prelude Manager (Concentrator)
(Gathers and persists events)
Sensor *n* (...)
Snort plugin
Prelude LML
(Parses logs and generates into events)
Snort
System logs
Global event processing and correlation

Figure 5-14: The Prelude IDS platform

The **Simple Event Correlator (SEC)** [Vaarandi2013] is a tool for event correlation in the domains of log analysis, system monitoring, network and security management, among others. Unlike most other event correlation products which are heavyweight solutions, SEC is a compact and lightweight and platform-independent event correlator which runs as a single process, requiring no graphical environment, with moderate CPU and memory requirements.

SEC is written in Perl and works on any UNIX platform with standard Perl support, without dependencies on any other software. It has also been used on Windows systems, requiring CygWin Perl [CygWin].

SEC reads log data from files, named pipes, or standard input, matching lines with patterns (like regular expressions or Perl subroutines) for recognizing input events, correlates such events according to the rules in its configuration file(s). SEC can produce output by executing external programs (e.g., snmptrap or mail), by writing to files, by calling precompiled Perl subroutines, etc. Event correlation configuration is specified as rules which are stored in text files. Rules are applied to input events in the order they are defined in the configuration file. rule definitions can have the following parts:

- Event matching pattern
- Boolean context expression
- Operation description string
- Event correlation parameters
- Action(s) for producing output

In addition, results from pattern matching can be cached, in order to reuse them at later rules. All SEC patterns can be extended for multi-line matching, in order to monitor log files with messages spanning over several lines. When an input event matches a rule, SEC will check if there is already an event correlation operation running for this event. If the operation exists, it will receive the event for correlation; otherwise, SEC will start a new operation which will then get the event. A rule could start many operations that are running simultaneously, while each operation has only one parent rule that started it.

NodeBrain [Boeing2013] is an open source interpreter of a declarative rule-based language, written in C programming language. This tool provides the basis for constructing an application for monitoring and event correlation. It provides a core engine that does the event correlation; this core parses rules presented in a declarative language. In addition to the core, Nodebrain provides a set of optional modules that extend its functionality. This modular architecture makes this correlator very flexible and lightweight as, according to the needs, modules can be added or removed.

The declarative language syntax and semantics is very complete and allows expressing complex attack scenarios.

Bellow we present a brief description of the most useful NodeBrain modules [Boeing2013]:

- Audit module- log file monitoring
- Baseline - statistical anomaly detection
- Cache - Repetition and variation detection, allows detecting pattern of events
- Pipe - named pipe communication
- Servant - child process delegation, parent can communicate via stdin/stdout
- Snmptrap - MIB-Less SNMP Trap Monitoring
- String - string manipulation
- Syslog - remote syslog monitoring
- Translator - recognizes elements of text and translates them it into NodeBrain commands
- Tree - lookup tables, stores information in a simple tree structure in main memory
- Webster - web server interface

Combining some of the modules around the core a complete event correlator can be designed.

It should be noted that the list of event correlation tools is by no means complete. The list presented here shows three mature open source event correlators that can fulfil the requirements of a correlator in the CockpitCI infrastructure.

5.3.1.3 Correlation and data acquisition in the CockpitCI architecture

Without the means to analyse and extract information from the data provided by the security probes, the proposed probing architecture from the previous chapter would be of little or no use. To such purpose, the probing layer is coupled to a distributed correlation infrastructure based on a two-level architecture that processes the event feeds provided by the sensors.

The two-level architecture increases the scalability of the correlation system, as this architecture allows overcoming the drawback of too many sensors sending events to a central correlator. This is achieved by having a level of correlators located in each one to the network zones and global one that receive events from the firsts.

This architecture is detailed in Figure 5-11 and encompasses the following components:

- Local level correlation is performed within each zone, using local correlators that may also have decision and reaction abilities, limited to the scope of its zone. Local instances report information to the higher level of correlation, also performing event reduction and synthesis (for instance, using duplicate elimination). This correlator acts like a data supplier for the main correlator. By performing event reduction and aggregation they send fewer events to the main correlator than those received from all the detections agents.

- The main correlator, which is placed above the local instances, gets a global perspective of the whole SCADA infrastructure, by receiving events from each local zone correlator. Due to the broad view of the whole infrastructure, this correlator has an important role in detecting network transversal attacks. This type of attacks happens when an attacker penetrates successive networks layers, starting in the IT network and progressing to Filed/Control Network. As the main correlator does not receive and need to process the events from all the agents in the infrastructure it increases the scalability of the architecture.

Also, there is the possibility of not only performing zone-related correlation but also service or device specific correlation, which is useful to deal with cyber-attacks targeting specific components. This correlation can be done at local level by the agent before sending the events to the local correlator because these rules can be very specific to the device or service. This allows reduction of events in the correlation layer as not all events are relevant to the correlators.

5.3.2 Machine learning mechanisms

As part of the CockpitCI intrusion detection strategies (and also included within the main PIDS correlation and analysis layer of the proposed architecture, in Figure 5-11), intelligent mechanisms based on machine learning and pattern recognition techniques (such as One Class Support Vector Machines) are also considered. Therefore, the inclusion of machine learning mechanisms will provide the means to gather knowledge about new data and make predictions about the future trends based on previously acquired information (from the previous data). This makes machine-learning techniques very useful for intrusion detection, to detect symptoms of rogue attacks without specific knowledge of their details.

Traditional signature and pattern-based systems are still very effective at detecting known attacks without generating an overwhelming number of false alarms. These techniques can quickly identify the use of a specific tool or technique. This can be very useful for security managers to prioritize the corrective measures and track security problems on their systems. To improve the range of analysis, in order to be able to detect rogue attacks, the cyber detection and analysis layer combines the rule-based approach a machine-learning module, further described in deliverable D3.2.

5.3.3 Security management platform and policy management

The security management platform is responsible for managing all the involved components of the solution. It includes the mechanisms for managing the security and components of the infrastructure.

While in the original architecture the Security Management Platform primarily addressed the maintenance and management of network monitoring devices such as IDS, it was later expanded to include monitoring of in-place security and vulnerabilities within the network as well as the maintenance of the latter. Therefore, this creates a distinction within the security management platform between security audit and maintenance mechanisms.

A policy management console can be integrated to provide a generic model of responsibility of components for cyber-attacks detection and strategic reaction to incidents. The policy management console can also be used to specify policies related to the detection, correlation and reaction.

All data regarding the detection system flows in a network that is separated from the SCADA network. This way, the traffic on the SCADA network isn’t affected by the detection traffic.

As such, the key features of the Security management platform can be summarised as:

- A console for appreciating both the security correctness and effectiveness of each of the security mechanisms (including IDS and honeypots) deployed within the infrastructure. Accounting for both the correctness and effectiveness of the security mechanisms within the detection architect has the merit to ensure that those mechanisms can be continuously updated to address newly emerging security threats and exploits.
- A console for gaining an insight of the status of the Security Relevant Components (SRC) within the infrastructure. An SRC being one that is not a security mechanism but when compromised can jeopardised the security of the entire infrastructure or the service that is being provisioned. An RTU is an example of such components.
- A policy management console whereby the administrator can define the set of rules relevant for detecting and containing anomalous activities.
    - By taking into account both the possibility of anomalies within the security mechanisms and within the actual network traffic and key components, the CockpitCI architecture would allow one, through a well defined correlation mechanism to determine any link between a network state of insecurity and the status of the security mechanisms.

5.3.3.1 Security monitoring and Maintenance Console

A number of recent studies such on security assurance [Ouedraogo2012; Ouedraogo2013] have revealed that a common but sometimes overlooked source of security risks for large distributed and open IT systems is the improper deployment of the security mechanisms. In fact the security mechanisms, even properly elucidated during the risks assessment stage, may be deployed inappropriately or unidentified hazards in the system environment may render them less effective. In fact, how good, for instance, is a fortified door if the owner, inadvertently, leaves it unlocked? Or considering a more technical example, how relevant is a firewall for a critical system linked to the Internet if it is configured to allow any incoming packets? Therefore, monitoring and reporting on the security posture of IT systems must be carried out to determine compliance with security policy [Jansen2009] and to gain assurance as to their ability to protect system assets and ensure business continuity.

With this is mind, a security monitoring and maintenance console ensure the management of the software probes involved in the verification of the security display a number of key metrics that could be used to make an informed decision on the status o those mechanisms and the ensuing alerts.

5.3.3.2 Network (SRC) Monitoring Console

Unlike the security monitoring and maintenance console, the Network monitoring console manages focuses on the actual network components and the traffic with the purpose to identify malicious traffics and potential vulnerabilities. It is the component whereby the configuration of IDS and other networks scanning tools such as Nagios will be performed. The console also interacts with the policy management console for determining the features of ingress and egress traffics to monitor. It is important to point out that the concept of correctness can be applied to a SRC to help determine whether its current status is conducive of the overall network /infrastructure being secure. The only caveat is that this technique may not be effective in case of the SRC component is compromise through a hijack of the communication links. Accounting for such a risk requires the consideration of the input/output of SRC. For instance, this will imply, establishing whether the relation between the set of instructions emanating the SCADA system square up with those emanating the RTU.

The key features of the network monitoring console include the configuration and set up of the available network monitoring tools based on a number of information emanating the the Policy management console. The nature of the rules set adopted for identifying any abnormal activities within the network would depend on a number of aspects including the

5.3.3.3 Auditing, vulnerability mapping and network monitoring

The identification of existing vulnerabilities within a network system is a big step towards staving off potential cyber-attacks that can be perpetrated against the system. Given the high complexity of today’s systems, and the rapid emergence of those vulnerabilities, vulnerability tools have been perceived as an efficient tool in the detection and prevention of cyber-attacks. Unlike Network Intrusion Detection Systems (NIDS), which are concerned with providing on the spot information on whether an attack is actually taking place, Vulnerability tools have a more proactive role. Indeed the precept of vulnerability tools is to gather and then convey to the network administrator the body of information relating potential flaws within the network that can be exploited in the future. Two types of vulnerability assessment are often used: (i) a host-based assessment of vulnerabilities that requires the actual software to be installed on a single machine with the aim to detect system-level vulnerability, and (ii) the network level assessment that is broader in the scope of its monitoring. Indeed this latest type of assessment is able to scour the whole network for identifying running services and the vulnerabilities that may be associated to them.

It is also useful a vulnerability analysis tool to test whether the servers, hosts, routers, and devices that are part of the SCADA network are vulnerable to known attacks. This tool performs host/network vulnerability analysis periodically (through port scanning and other mechanisms) and provides a visual map of the vulnerability that alerts the operators/engineers to take appropriate remedial actions. The tool has to be flexible so that new attacks can be added to the repertoire any time. The tool acts as a security management technique, and complements the IDS techniques. Examples of such tools are the Nessus [Nessus], Metasploit [Metasploit], Core Impact [CoreImpact] and Canvas [Immunity] modular penetration-testing frameworks for which SCADA modules are available (for instance, [SCADAHacker] lists several modules for Metasploit). Specifically, Metasploit is one of the most widely used frameworks, being part of almost any security expert and penetration tester toolkit. By encompassing many different capabilities and components, Metasploit can be used for a wide range of tasks from penetration testing to check if a given server has updated operating system patches installed. These components will be discussed in further detail in deliverable D3.4.

Even if these tools can be used for legitimate purposes, it must not be forgotten that malicious usage by some attacker is also a possibility that cannot be ignored – in fact, there is no standard to distinguish a penetration testing tool from a hacking tool. As such, their usage must be part of the periodic internal security assessment routines. As these tools are continuously updated, sometimes with short development cycles, the assessment periodicity must be adjusted accordingly. The scope of penetration testing assessments can be established based on Service Level Agreements (which are even more useful, if an external security consultant is involved), in order to provide accurate results without putting critical systems at risk, therefore ensuring a balance between reliability and auditing precision.

5.4 Requirements for interface with the mediation network and the Risk Prediction Tool

The CockpitCI detection and analysis layer, which is part of a CI Distributed Perimeter Intrusion Detection System (PIDS – see Figure 5-3), must be able to somehow communicate its findings and related security events to the Secure Mediation Network (SMN - as discussed in further detail in deliverable D5.2), via the Secure Mediation Gateway (SMGW). This topology will enable the exchange of information between CIs, also enabling the use of risk prediction and other analysis mechanisms to assess threats in a global scale, accounting for CI interdependencies. These requirements, regarding the Secure Mediation Network interface (via SMGW), which also are specifically related with the cyber detection layer and which are discussed with further detail in deliverables D5.1, D5.2 and D5.3.

In this scope, SMGWs will be responsible for the exchange of prediction tool results, also supporting secure real-time exchange of alarms directly generated by the detection system. In this sense, this is one of the main elements of the CockpitCI tool, since all information exchanges are vehiculated through this component, including the information generated by the cyber detection layer.

Figure 5-15 (extracted from D5.3) shows a simplified description of the CockpitCI tool and its interconnection within the CI.

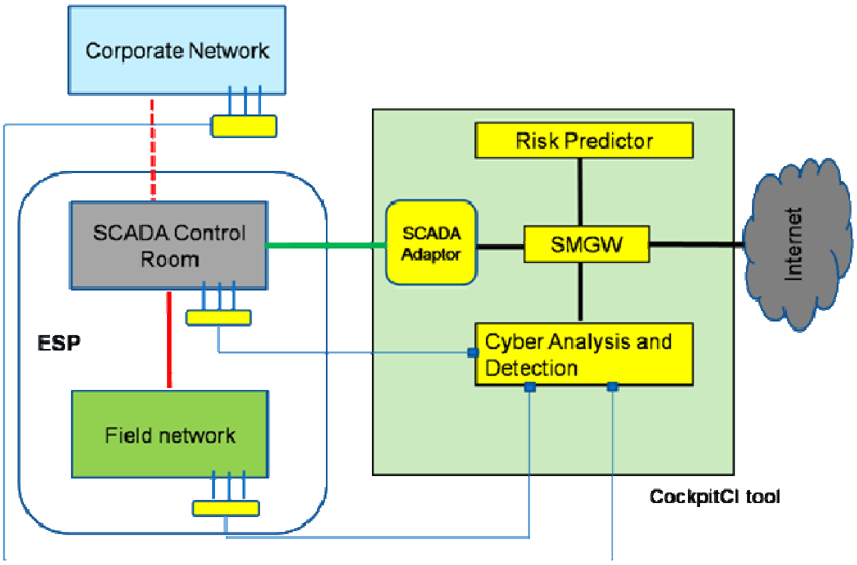


Figure 5-15: Schema of CockpitCI integrated within a CI (as per D5.3).

As previously described, the CI includes three fundamental zones (Field Network, IT Network, SCADA Process/Operations Network – see Figure 5-11), that constitute the Electronic Security Perimeter (ESP) which must be continuously assessed and protected.

As shown in the Figure 5-15, there are three logical interfaces to the SMGW, for each CI:

- Interface with the SCADA adaptor (described in deliverable D4.1) to extract relevant SCADA information from the SCADA control room (via HMIs, for instance).
- One for the cyber detection layer, which is relevant in the scope of this deliverable.
- And one for the Integrated Risk Predictor (IRP) for that CI.

Regarding the interface between the cyber detection layer and the SMGW, it must be taken in to account all the needed precautions to avoid creating a new attack vector. Therefore, the Cyber Analysis and Detection Layer must adopting state-of-the-art security solutions to block propagation of cyber threats - these solutions range from passive probes to the use of secure protocols.

Since CockpitCI proposes to simplify and rationalize the original MICIE SMGW solution, it needs to improve scalability while being able to the deal with the increased complexity of the CockpitCI scenario. Therefore, the cyber detection layer will have to be enhanced with coupling agents to provide the interface with the SMGW, eventually using a client/server model based on the use of web-service technologies.

Also, it might be desirable to switch the locations for event/information buffering from the SMGW to the cyber detection layer. It would be probably more efficient from a network communications standpoint (less message exchanges, less RPC calls, since the communications between the detection layer or/and the SMGW would be request-driven and not even-driven), but with a penalty on latency and on cyber detection event publishing (because of the buffering/memory requirements).

Considering the desirable IRP prediction cycle latency for CockpitCI (which is around 10 sec), this could be a possibility. In MICIE the working test-bed provided a different time cycle for each element (each adaptor and IRP). So from IRP point of view, only the latest values were requested, without the need for buffers.

5.5 Domain-specific components

This section describes the components having specific behaviour regarding the domain where they are deployed.

5.5.1 Field Security Manager (FSM)

The Field Security Manager is a system that hosts the security mechanisms for a given Autonomous System (AS). This system hosts the local correlator, the Backup Master Station and Heartbeat logic for a field network classified as AS. This local correlator processes events from the AS NIDS, but also from the Shadow RTUs.

This approach goes in line with the Smart RTU layer, in a sense that the FSM is a self-contained entity for an AS, designed to ensure autonomous capabilities in case of unexpected failure or explicit operation causing isolation of the AS.

5.5.1.1 Backup Master Station (BMS)

A Backup Master Station consists in a local master station on a given Field Network, i.e., an AS, hosted by a Field Security Manager (FSM) system. All Field networks have their separate BMS, providing some level of autonomy in the event of becoming isolated from the SCADA system.

The BMS can coordinate the RTUs to perform pre-determined actions like an orderly shutdown, also providing self-healing and recovery procedures (for instance, in case of an attack reprogramming RTUs or PLCs, the control center might force the AS into isolation so that the BMS might reprogram and refresh the file unit code).

5.5.2 Heartbeat mechanisms

Some components need to know when they are isolated from the system, e.g., correlators, BMS. To accomplish that they use a heartbeat mechanism, consisting of sending a periodic signal and receiving its response. When they fail to receive a signal response, they can assume an isolation scenario and take the pre-determined actions. Once again, this mechanism is complementary to the Smart RTU layer, in the sense that it might be used to provide (semi-) autonomous capabilities to an AS on the field. The heartbeat mechanism is also useful to the main correlator in the sense that if an AS becomes isolates, actions may be taken to the remaining system.

Similarly, embedded watchdog mechanisms might be evaluated as a complementary alternative to provide an added protection layer providing self-healing and recovery capabilities.

5.6 Other security countermeasures

The following sections describe other relevant security topics for the critical infrastructure topics. Although they are not part of the detection architecture proposed in this document, they should be considered for other parts of a complete architecture such as the reaction mechanisms or maintenance approach.

5.6.1 Policies

The foundation of any effective cyber security program is the cyber security policy. Although, they can range in size and style, there are usually several themes that are always mentioned. A standard cyber security policy can include [GAO2005]:

- Policy upkeep, refinement of policy, and compliance.
- Cyber security countermeasures.
- Cyber security technologies.
- Incident response.

For the scope of the CockpitCI detection architecture is important to mention the policies and approach used to update the detection agents. For instance the effectiveness of an NIDS rule based relies on continuous update of its rules, for instance in a daily fashion. Other common topics such enforcing physical and proper user access control is also decisive since they are one of the common sources of attacks and should therefore not be neglected.

5.6.2 Antivirus/antimalware

Contemporary ICT security systems are often deployed with countermeasures to mitigate virus, malicious software, and other types of malicious code have some sort of transport capability of are used specifically to increase an attacker's level of compromise [GAO2005]. Implementing antivirus and malware protection on critical systems can help detecting and defeating such attempts, not only for viruses, malware and worms but also for malicious activity as well, being able to detect hacking tools. Any notification of these products must be logged to a centralized sever, with notifications being sent to administrators. There is a concern about the way anti-virus might affect the real-time performance of critical control systems (such as Master stations). As individual mileage may vary, a previous assessment of the antivirus or malware protection performance overhead may be advisable.

They are used to detect malicious code under a host and although they add a small overhead to the system, they should be considered as a useful input of events for specific attack scenarios. For instance an network probe using forged addresses using an piece of code inside a SCADA component can be traced using those types of detection engines. Notice that because of forged addresses an NIDS may not be sufficient to trace the attack source.

5.6.3 Firewalls and network perimeter separation

Most people understand the principle of the firewall and how they provide security [GAO2005].

Firewalls work in much the same way that burglar alarms or anti-tamper technology can be used to detect and thwart attack attempts. Intrusion detection and intrusion prevention (IDS and IPS) are used as alarm mechanisms to indicate possible malicious activity, technically, are two different security solutions [GAO2005].

Since the most important threat to the SCADA network may come from malicious attackers via the Internet, it is necessary to monitor the traffic flows from the Internet (IP network) to the SCADA network. Generally, firewalls and other Intrusion Detection Systems (IDS) are installed at the various ingress points (gateways) of the SCADA network to identify malicious traffic before it is allowed to enter. Although this would help to filter out some attacks, it may still be an inadequate defence action against attacks. Viruses and worms could swamp the systems with huge volumes of attack traffic. Hence, having only firewalls and IDS at entry points may not suffice. This leads to the concept of the electronic perimeter.

It is proposed that a wider electronic perimeter be defined where cyber attacks can be filtered and unwanted traffic stopped before it reaches the SCADA network gateway.

The extended perimeter can be formed by multiple IDS devices across a wide area. Huge volumes of traffic can be handled by an extended perimeter as it would be possible to stop the attacks further away from the SCADA network. In addition, the IDS devices along the electronic perimeter could form an overlay network (i.e., a virtual private network over the Internet) and function in a distributed and collaborative fashion, supporting one another in tackling the attacks more effectively. The setup can be viewed as an electronic fence or protective perimeter barrier that allows only legitimate traffic to reach the gateway of the SCADA network.

In the CockpitCI project is proposed an multi-zone approach having an IDS between each one. Nevertheless, they are limited to detection role and therefore, additional reaction mechanisms such the use of a firewall is advisable. For instance an NIDS running in a passive mode, as proposed in this architecture, may detect a network attack but cannot block it.

Among the topologies to enforce separation between the ICT and SCADA networks, [Byres2005] discusses three main techniques:

- Dual-homed computers
- Two-zone separation
- Multi-zone separation, with a DMZ

The first architecture is the most simple way to separate networks, by using multi-homed systems (hosts with two network adapters, one placed in each network) in all systems that need access to both the ICT and SCADA network (see Figure 5-16) .

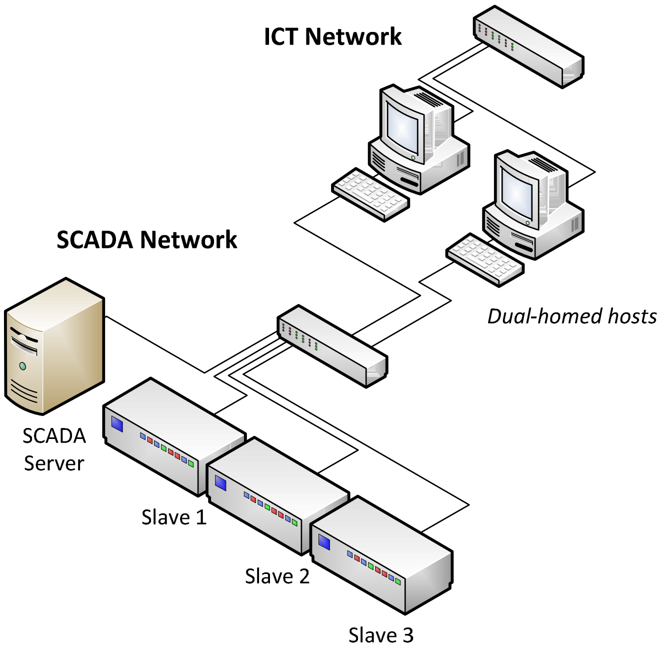


Figure 5-16: Network separation using dual-homed systems

Albeit simple, this approach poses a serious security risk as it doesn't enforce any kind of restriction *per se*, once an attacker gains control of one of the multi-homed hosts. However, a survey by [Byres2005] found that several organizations were using this topology.

Topologies based in the (two-zone) separation of the ICT and SCADA networks (see Figure 5-17), by means of a firewall/router are a significant improvement in terms of security.

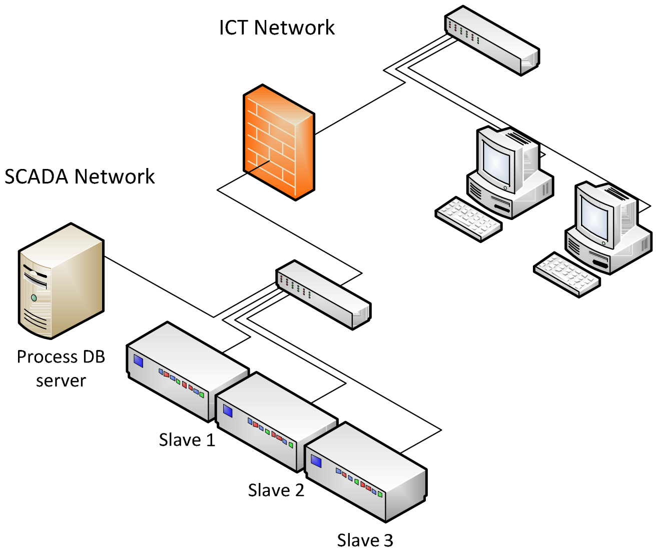


Figure 5-17: ICT and SCADA network separation

While they offer increased security, this simple separation means that, in the case a specific host or server is of use for both networks, the firewall must be pierced to enable direct network traffic flows between them, a situation that poses a security risk.

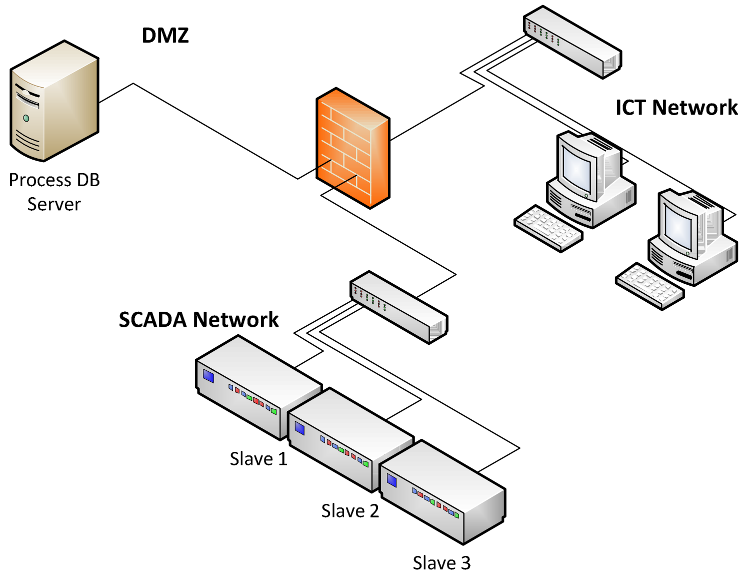


Figure 5-18: Multi-zone topology with a DMZ

The third alternative (see Figure 5-18) is based on a multi-zone topology [NISCC2005], in which the ICT and SCADA networks are joined by a DMZ (an acronym which means “De-Militarized Zone”), were all systems that must be shared between both networks are placed.

5.6.4 Penetration testing

Penetration tests have been routinely used for evaluating the security of computer systems or networks by simulating an attack from a malicious source. The process involves an active analysis of the system for any potential vulnerabilities that may result from poor or improper system configuration, known and/or unknown hardware or software flaws, or operational weaknesses in process or technical countermeasures. For this, the testing team would be engage first in the identification and construction of scenarios as well as dependency and security attack. Security test cases are subsequently defined, from those scenarios to test the developed design of the system against various types of attacks.

The IT penetration test establishes how far an attacker could penetrate the system. Therefore, the organisation employees are often unaware of the testing to prevent biased results. On the other hand, when asset owners prescribe a cyber security assessment of an ICS, they want to know if vulnerabilities exist inside the hardware and software that make up the ICS and whether the protections (network architecture, functional DMZs, sensors) in place will limit access. Corrective actions can be taking as a result to mitigate any future real attack on the infrastructure.

The downside of Penetration testing lies on its potential to disrupt normal activities and worst, result in some case in the damage of some components.

5.6.5 Secure communications

Secure communications are usually referred as additional security mechanisms to protect plain text communications and access from an attacker. A plain text communication can be eavesdropped, manipulated or totally forged. In the same way, a system or service without proper access control can be easily used and abused. In the detection architecture this means the possible to forge alarm events, suppress them by Man-In-The Middle (MITM) attacks or simply compromise the detecting agents themselves.

This section encompasses three main topics: communication flow security, communication service security and network equipment security.

- The various communication links must be secured by adopting well known security standards such as VPN and Internet Protocol Security (IPSec) to provide authentication, data integrity and confidentiality for the data communication between the Internet or corporate network and the SCADA network. However, special care must be taken to ensure that the latency overhead of using strong encryption for real-time communications doesn’t impair the communication, especially if control streams are involved.
- Also, DNS Security (DNSSEC) must be deployed in all DNS servers associated with the electric grid for validating the authentication and the integrity of DNS transactions. The use of the Dynamic Host Configuration (DHCP) should be avoided (or, at least, closely monitored and strictly managed using static leases), whenever possible, keeping the infrastructure as static as possible. Also, it is good practice to completely disable all network services that aren’t being used or others with a poor security record (such as the Server Message Block or Universal Plug and Play services, for instance), therefore streamlining the protocol and service ecosystem to the bare minimum, with the added benefit of reducing entropy and easing monitoring.
- As for network equipment, port-based access control on wired networks, using 802.1X, may be provided to extend the reach of RBAC (Role-Based Access Control) and AAA mechanisms into the network equipment (such as switches), providing an added access control layer.

  Spanning tree protocol (STP) attacks, may provide an attacker with physical access to the network to create BPDU (Bridge Protocol Data Unit) frames to produce a deviant behaviour of the STP on switches, enabling network disruption (for DoS purposes), or even worse, to take control of the root bridge for MITM attacks (in which the fake root bridge makes itself part of the critical path for network traffic, disabling all other paths). In SCADA networks, STP must be disabled if possible – in alternative there are technologies such as Cisco root guard and BPDU guard that allow enforcing a perimeter around a protected network to protect it from attacks. These features can be enabled on a per-port basis, enabling consistency verification of STP operations and subsequent alerting.

ARP cache poisoning is another technique that can be used to subvert switch operation, providing the basis for MITM attacks. Use of port security (statically locking switch ports to MAC addresses) and low-level detection techniques such as ARP probes (using Wireshark [Wireshark] or Arpwatch [Arpwatch]) it is possible to detect and deal with these attacks. Also, there are mitigation techniques that are particular of each equipment manufacturer, such as Cisco's DAI (Dynamic ARP Inspection).

5.6.6 Users/employees training of security

Shaw [2013] remarked that erecting a secure cyber-barrier around your SCADA system is a good idea and not an insignificant effort though many of the ways in which a SCADA system could be disabled, damaged or used to wreak havoc, would involve an "inside job". That is, the human aspect of the security of ICS should not be underestimated. Indeed no matter how strict a security mechanism is; the security of a system would still depend on the users' security train.

The need for convenience by users often conflict with security imperatives to such an extent that end-users would often bypass security controls in order to fulfil their tasks. This is often translated into the selection of weak and easy to remember authentication such as passwords, which could also be shared amongst colleagues or simply written down.

As a result employees involved in CI ought to receive frequent and up to date training on the risk that they may inadvertently pose by something not being rigorous in their implementation of the security policy at their working place. More importantly each employee should be trained and exposed to the latest Social engineering technique.

5.7 Cyber-attack detection and communication

If the detection of abnormal events in the overall systems of a CI is the first step to ensuring sustainable risk management, the second step is the communication of relevant information to other systems and users (incident response team, operational level, management).

A number of different taxonomies have been established during the last decade [Simmons2009] to formally describe a cyber-attack. The following taxonomy is based on AVOIDIT (Attack Vector, Operational Impact, Defense, Information Impact, and Target) and has been completed by adding supplementary categories in order to be able to precisely define cyber-attacks targeting both SCADA and IT networks. The proposed taxonomy is also based on a paper focusing on the classification of Cyber-attacks on SCADA systems.

5.7.1 Taxonomy description

Figure 5-19 provides an overview of the taxonomy. The taxonomy classifies cyber-attacks from 5 points of view: "*Attack vector*", "*Operational Impact*", "*Defence*", "*Informational Impact*" and "*Attack Target*". As mentioned by the author of the AVOIDIT taxonomy, the logical requirements of the classification are the following:

1. *Mutually exclusive: each attack can only be classified into one category, which prevents overlapping.* Even if a complex attack has to be classified in more than one type of cyber-attack according to the level of the exploit or depth of the attack.

Table 5-1: Attack description example

ID	Parent	Name	Attack Vector	Operational Impact	Defense	Informational Impact	Target
001	-	Industrial spying 1	Social Engineering	User Compromise	Awareness	Disclosure	User
002	001	Industrial spying 2 …	Social Engineering	Misuse of resources	Awareness	Disclosure	Local

-

2. Comprehensible: Clear and concise information; able to be understood by experts as well as those who are less familiar.
3. Complete/Exhaustive: available categories are exhaustive within each classification, it is assumed to be complete.
4. Unambiguous: involves clearly defined classes, with no doubt of which class the attack belongs to.
5. Repeatable: the classification of attack should be repeatable.
6. Terms well defined: categories should be well defined, and those terms should consist of established terminology that is compliant within the security community.
7. Useful: the ability to be used to gain insight into a particular field of study, particularly by those having great interest within the field of study.

The AVOIDIT taxonomy paper, mentioned above, gives a precise description of the chosen categories which can be summarised and completed as follows:

Attack Vector: An attack vector is defined as a **path** by which an attacker can gain access to a host. The majority of attacks can be described according to the following attack vector:

1. Misconfiguration: use of a configuration flaw within a particular application to gain access to a network or personal computer.
2. Kernel Flaws: use of a kernel flaw within an operating system to gain certain privileges to exploit vulnerabilities within the operating system. (e.g. Vulnerability on kernel of Wind River System VxWorks which is used in hundreds of devices: http://www.kb.cert.org/vuls/id/362332)
3. Design Flaws: use of a design flaw within a system or device to retrieve sensitive information (e.g. password theft using the firewire or thunderbolt connectivity flaw

even if the computer is locked and encrypted: http://erratasec.blogspot.com/2011/02/thunderbolt-introducing-new-way-to-hack.html#.UZ-iOypXvJE)

4. Buffer Overflow:Buffer overflow is caused when a piece of code does not adequately check for appropriate input length and the input value is not the size the program expects. An attack can exploit a buffer overflow vulnerability leading to a possible exploitation of arbitrary code execution.
5. Insufficient Input Validation: A program fails to validate the input sent from a user. An attacker can exploit the insufficient input validation vulnerability and inject arbitrary code (e.g. SQL injection).
6. Symbolic Links: A file that points to another file. An attacker can exploit a symbolic link vulnerability to point to a target file for which an operating system process has write permissions.
7. File Descriptor: A file that uses numbers from a system to keep track of files, as opposed to file names. Exploitation of the file descriptor vulnerability allows an attacker the possibility of gaining elevated privileges to program related files.
8. Race Condition: Occurs when a program attempts to run a process and the object changes concurrently between repeated references allowing an attacker to gain elevated privileges while a program or process is in privilege mode.
9. Incorrect File/Directory Permission: An incorrect permission associated to a file or directory consists of not assigning users and processes appropriately.
10. Social Engineering: The process of using social interactions to acquire information about a victim or computer system, which, in normal circumstances, is not available.(e.g. phishing is a social engineering method to penetrate systems, even those protected by technical systems like IDS: http://www.social-engineer.org/framework/Real_World_Social_Engineering_Examples:_Phishing)

Operational Impact: An operational impact is defined here as an evaluated consequence of an attack at operational level (IT and SCADA level). Classification by Operational Impact involves the ability for an attack to culminate and provide high level information known by security experts, as well those less familiar with cyber-attacks.

1. Misuse of Resources: An unauthorised use of IT/SCADA resources or IT/SCADA functions (usable with specific privileges).
2. User Compromise: Gaining unauthorised use of user privileges on a host.
3. Root Compromise: Gaining or elevating privileges to unauthorised privileges of an administrator on a particular host/ system.
4. Web Compromise: A website or web application using vulnerabilities to further an attack (cross site scripting or SQL injection).
5. Installed Malware: An attack can be launched via user installed malware, whether by intentional installation or drive-by installation. Installed malware can allow an adversary to gain full control of the compromised system leading to the exposure of sensitive information or remote control of the host.

   a. Virus– A piece of code that will attach itself through some form of infected files, which will self-replicate upon execution of a program. (boot record infectors, file infectors, and macros).
   b. Spyware: collecting information from a computing system without the owner's consent.
   c. Trojan: A benign program that allows unauthorised backdoor access to a compromised system.
   d. Worms: A self-replicating computer program that spreads throughout a network. Worms include mass mailing and network aware worms.
   e. Arbitrary Code Execution: Involves a malicious entity that gains control through injecting its own code in order to perform any operation on the targeted application.

6. Denial of Service: Denial of Service (DoS) is an attack which denies a victim access to a particular resource or service i.e.:
   a. Host Based: A Host based DoS aims at attacking a specific computer target within the configuration, operating system, or software of a host. These types of attacks usually involve resource hogs, aimed at consuming up all resources on a computer; or crashers, which attempt to crash the host system.
   b. Network Based: A Network based DoS targets a complete network of computers to prevent the network from providing normal service. Network based DoS usually occurs in the form of flooding with packets, where the network's connectivity and bandwidth are the target.
   c. Distributed: A Distributed Denial of Service (DDoS) is becoming increasingly more popular as an attacker's choice of DoS. A distributed denial of service uses multiple attack vectors to obtain its goal.
7. Timeliness degradation: This attack aims to stop a system responding on time to commands. This type of attack will degrade the Quality of Service (QoS) provided by a system by targeting either the entire system, specific system functionality or system resources and can seriously impact the QoS of a Critical Infrastructure if it targets industrial components such as a PLC controller. The timeliness aspect includes both the responsiveness of a system (real-time response) and the freshness of data (for an industrial system, the data is only valid in a designed time period).

Defence: Classification by defence highlights several strategies a defender can employ to remain vigilant in defending against pre and post attacks.

8. Mitigation: A form of defence used prior to vulnerability exploitation or during an attack, to mitigate damage an attack has caused, or has the potential to cause. Mitigation involves reducing the severity of the attack.

   a. Remove from Network: The ability of an administrator to remove infected hosts, thus preventing further damage.
   b. Whitelisting: A list of permissible connections that are known to the defender.
   c. Reference Advisement: Notes provided by the defender to mitigate an attack, or a vulnerability/vendor database reference number used to alleviate a vulnerability or attack.
   d. Awareness

9. Remediation: Defence used, in the presence or prior to vulnerability exploitation, to prevent an attack.
   a. Patch System: Applying patches which have been released due to software vulnerabilities.
   b. Correct Code: Steps within an organisation to release a code patch to a specific application that will eliminate the potential for an attacker to exploit.
   c. Shielding: Steps within an organisation to avoid unnecessary physical or logical access to system resources.
   d. Replacement: Steps within an organisation to remove the out-dated system or breakdown system and to replace it with a more secure system.

Informational Impact: An informational impact is defined here as an evaluated consequence of an attack on the reliability of information used (confidentiality, integrity and availability of information) at operational level (IT and SCADA level). An attack on a targeted system has the potential to impact sensitive information in various ways. A committed resource must be able defend information warfare strategies in an effort to protect themselves against theft, disruption, distortion, denial of service, or destruction of sensitive information assets.

10. Distort: A distortion of information, usually when an attack has caused the modification of a file.
11. Disrupt: A disruption to services, usually from a Denial of Service attack, involving unavailability of information access.
12. Destruct: A destruction of information, usually when an attack has caused a deletion of files or a removal of access.
13. Disclosure: A disclosure of information, usually providing an attacker with access to information that they would not normally have access to.
14. Discovery: To discover previously unknown information. For example, when a scanning tool probes for information, the information discovered can be used to launch an attack on a particular target.

Target: Generally an attack targets a specific type of host. The classification assigned to the target is able to improve the defence of a whole set of systems by adapting mitigation and remediation actions to a specific range of systems.

15. Operating System (Kernel / User / Driver): Responsible for the coordination of activities and distributing the resources of a computer. An attack can be designed to target vulnerabilities within a particular operating system which can be defined by its family (Microsoft Windows), its name (e.g. MSWIN7), and its version (MS Windows 7 64-bit SP1).
16. Network: To target a particular network or gain access through vulnerability within a network or one of the network protocols. The network target can be specified by its Area (Corporate/Enterprise network, IT operational network, SCADA network etc…) its Type (wired, wireless, radio waves etc…), its Protocol (ModBus, Ethernet [802.3], internet [IPV4], Synchronous Optical Network (SONET) [OC1] etc…) and its Version.
17. Local: An attack targeting a user's local computer.
18. User: An attack against a user is an attack to retrieve a user's personal information.
19. Application: An attack towards specific software. An application can be either client or server. A client application is software that helps a user perform common tasks. A server application is software designed to serve as a host to multiple concurrent users.

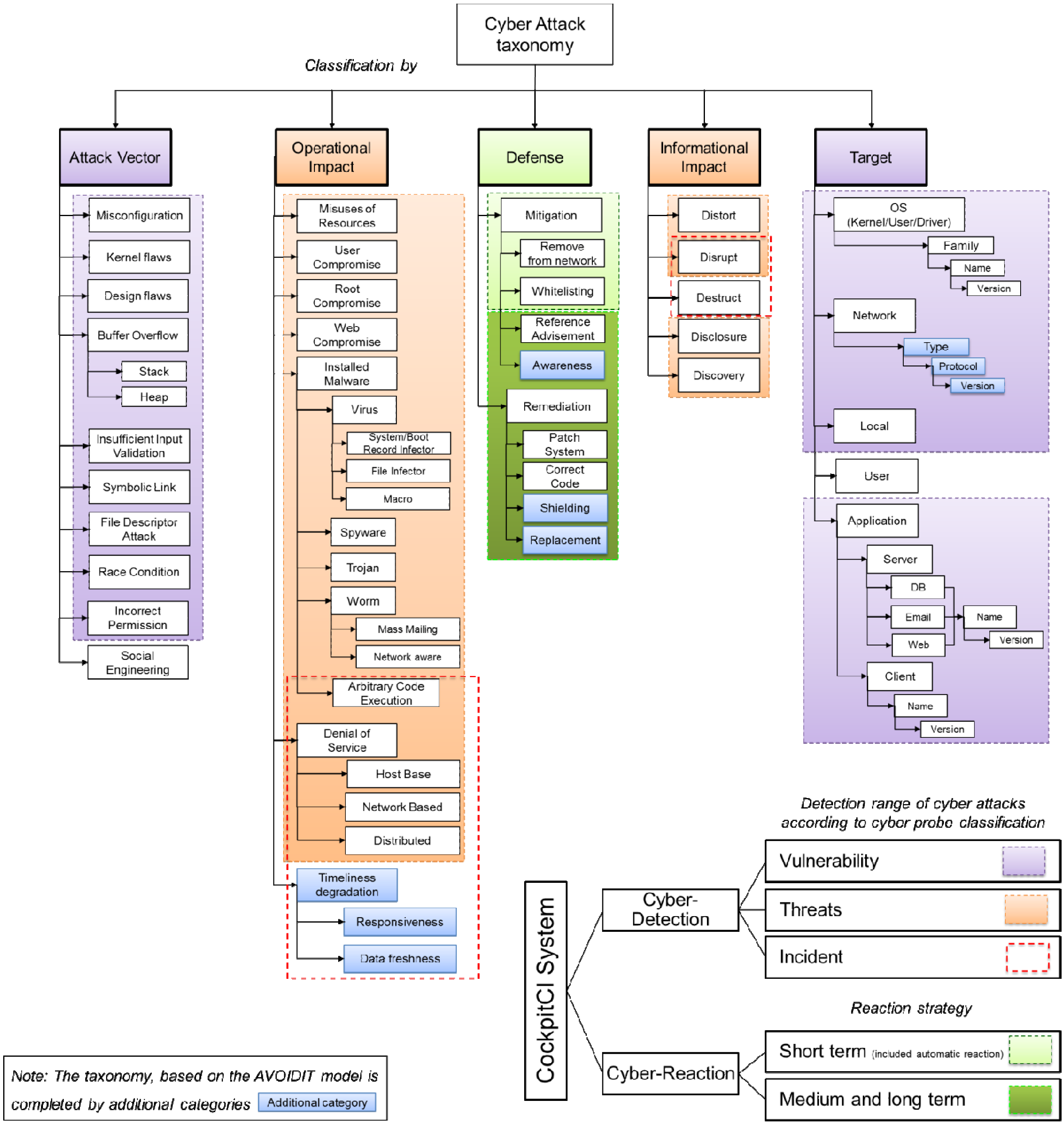


Figure 5-19: Cyber-attacks taxonomy for CockpitCI system

5.7.2 Cyber-attacks taxonomy and the cyber detection and correlation layers

The full description of an attack according to the 5 classification categories mentioned above is almost never possible in real-time. The full description always involves an in-depth analysis of the attacks by the means of a forensic analysis or a code analysis (especially to suitably design the method of defence). However, within the framework of the CockpitCI system, this taxonomy could be successfully used to set up a meta-language which could be used to retrieve consolidated and understandable information from detection layer to high level (RPT and simulator) and assess the risk level of the overall infrastructure. With that aim in mind, Figure 5-19 also describes the link between each type of category and the information retrievable from the field by cyber detection agents.

- The detection agent targeting the vulnerability (e.g. vulnerability checker) could be used to retrieve information on the attack vector and target;
- The detection agent who is able to identify threats could be used to retrieve information on operational and informational impacts;
- The detection agent recording security and operational incidents could be used to retrieve information on operational and informational impacts.

Note: some information retrieved by the detection layer can be naturally correlated. For example, the detection of a malware is generally made on a specific network and before any analysis takes place all hosts connected to that network can be considered as a potential target. The refinement on the nature of the malware provided both by the detection agents (e.g. crash of a specific system) and malware analysis (online and offline) will allow specific hosts to be targeted.

During the overall attack process, the detection layer; including the detection agents, local and main correlators and analysis system will provide more and more information to increase or decrease the risk level of components included in the overall CI's architecture according to the effectiveness and the range of a specific attack. For example:

- The detection layer provides an overview of the vulnerable system (vulnerability checker): x systems are not correctly patched. These x systems are vulnerable and the risk level of the overall infrastructure is increased to level 1 according to this information. Following this, the detection layer provides information that a system y included in these x systems has crashed without reason. Therefore, we can justifiably increase the risk for all systems similar to y to level 2.
- The detection layer identified an abnormal amount of traffic over a specific network (e.g. an unknown executable file). All systems able to run such exec files and connected to this network are potential targets: The risk level of the set of these systems will increase to a higher level e.g. 2. If for example we then identify the executable as a well-known malware classified according to the cyber-attack taxonomy, only systems that are potential targets will keep a high-risk level; the risk level of the other systems will decrease.

To better detail which mechanisms are most effective when dealing with specific threats, the Table 5-2 includes a simple security ontology. This ontology is by no means exhaustive, as its main purpose is to show the extent of the effectiveness of each security mechanism, in the scope of the proposed reference cyber-analysis and detection architecture for CockpitCI.

Table 5-2: Security ontology for the components of the CockpitCI cyber-analysis and detection reference architecture

Mechanisms	Cyber-threat/symptom	Reason
Shadow RTU	Sending of unauthorized commands or master impersonation	The shadow RTU is able to detect abnormal command activity from unauthorized origin.
	RTU reprogramming	The shadow RTU is able to detect abnormal behaviour from the monitored device.
	Abnormal delay	In some situations, the shadow RTU may be able to monitor traffic and detect excessive delay.
	MITM attacks	The shadow RTU is able to detect abnormal behaviour from the monitored device, by monitoring commands and actions.
	Probe attacks	The shadow RTU is able to detect abnormal command activity from unauthorized origin.
Honeypots	Probe attacks	The presence of traffic on the honeypot is a sign of unauthorized activity.
	Sending of unauthorized commands	The presence of traffic on the honeypot is a sign of unauthorized activity.
	Master impersonation	The presence of traffic on the honeypot is a sign of unauthorized activity.
	RTU reprogramming	The presence of traffic on the honeypot is a sign of unauthorized activity.
Network IDS	MITM attacks	Depending on the nature of the attack, Domain-specific NIDS can track state changes on the command flow.
	Probe attacks	NIDS are able to detect traffic traces corresponding to such situations.
	Sending of unauthorized commands	Domain-specific NIDS can monitor the command flow and detect these issues. However, conventional NIDS can also be useful in some cases, when the

		commands come from an unkown Master Station.
	Master impersonation	When properly configured, NIDS can detect abnormal command flows from unknown origins.
	IP protocol level attacks (Smurf, Address Resolution Protocol (ARP) spoofing, flooding, etc.)	When combined with firewalls, NIDS can be very effective in detecting and stopping such attacks
	DoS attacks	When combined with firewalls, NIDS can be very effective in detecting and stopping such attacks
Host IDS	Rootkits	HIDS are able to detect signature changes on critical system files.
	Tampering	HIDS are able to detect signature changes on critical system files or unexpected configuration changes.
	RAT attacks	HIDS are able to detect signature changes on critical system files, but can also check for open TCP ports or unexpected configuration changes.
	Worms and virus	HIDS are able to detect signature changes on critical system files or unexpected configuration changes
	Unauthorized access	HIDS are able to analyse logs to check for unauthorized access.
Field Security Manager/ Backup Master Stations	Abnormal delays and interruptions	FMS/BMS has the means to correlate information about abnormal behaviour on the AS, being able to (accordingly with established policies) to proceed with autonomous remediation.
	Unexpected systems isolation	FMS/BMS has the means to ensure safe operation levels are maintained (accordingly with established policies).
	Abnormal PLC behaviour	FMS/BMS has the means to initiate remediation actions in such cases (accordingly with established policies), which can go to the extent of reprogramming the field device.

MultiZone Correlation	Zone-specific abnormal activity	System is able to detect and pinpoint specific problems, affecting a particular CI zone.
	System-wide abnormal activity	Enables detection and tracking not only of ongoing threats, but also of *ab initio* symptoms, related to probing and attack preparation. Global correlation provides a broad perspective on the security status of the CI, enabling detection of sophisticated behaviour patterns, involving several zones of the CI.

5.7.3 Cyber-attacks taxonomy and risk analysis

The cyber-attack taxonomy allows us to perform a risk analysis compatible with the ISO 27005 standard, as the identification of attack is in line with the risk identification of the standard.

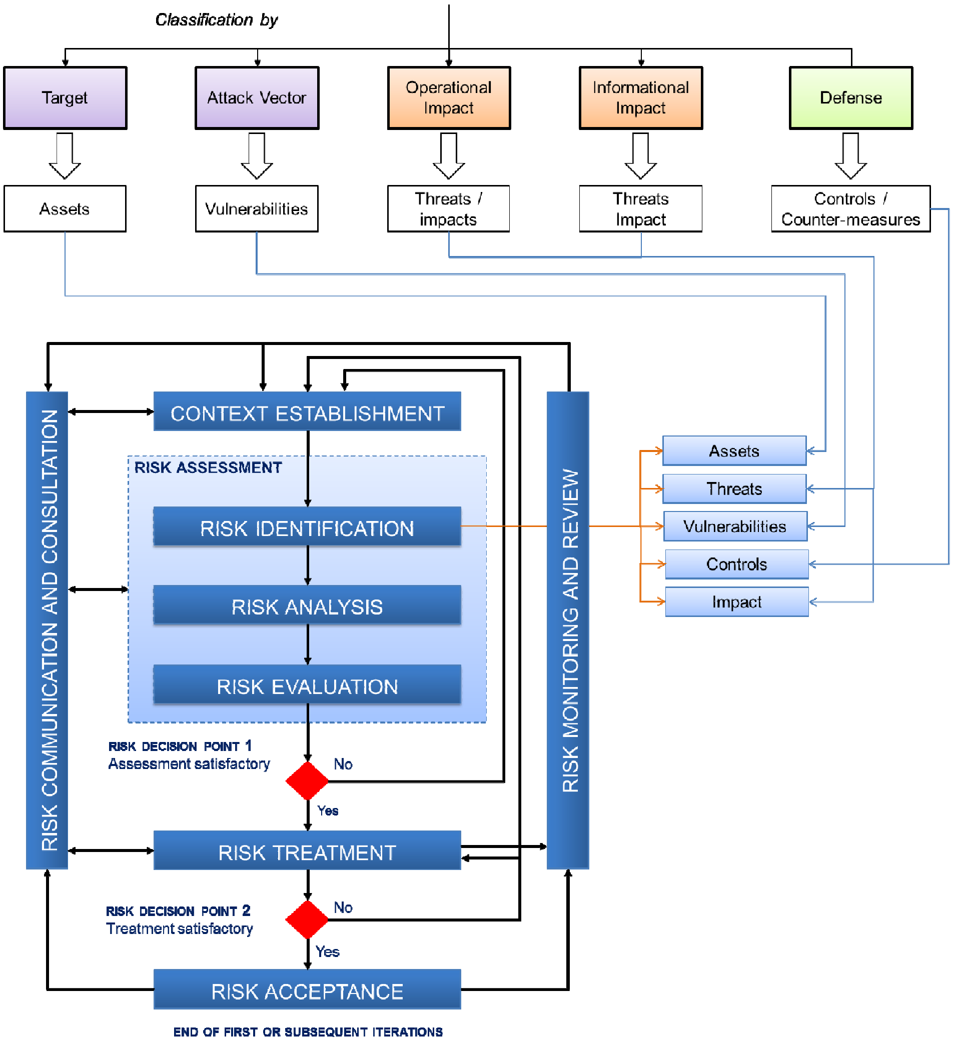


Figure 5-20: ISA 27005 (adapted from [ISA27005])

Note: According to the standard, the risk analysis aims at assessing the probability of a threat to successfully exploit vulnerability and to assess the impact of a successful attack according to a fixed scale. The risk evaluation has to assess the level of risk according to a fixed scale of risk level. In CockpitCI these steps of the risk analysis are performed by the cyber-simulation and by the prediction tool. The Incident Management Team will perform the risk treatment.

5.7.4 Application of taxonomy to CockpitCI use cases

This section provides a short presentation of the expected inputs of the Cyber-detection layer according to the previous taxonomy and the relationship between this taxonomy and the information retrieved from the detection layer. We apply the taxonomy to the first use case of CockpitCI project: Malware Spreading

The first use case considers the occurrence of a cyber-attack, which injects malware into a telecommunication network such as a SCADA or Enterprise network.

Injection of malware in a telecommunication network can use several types of attack vector. We propose the most common way i.e. Insufficient Input Validation, Misconfiguration and Design Flaw.

The identification of such cyber-attack according to our taxonomy is:

Table 5-3: Cyber attack identification

ID	Name	Attack Vector	Operational Impact	Defense	Informational Impact	Target
001A	USE CASE 1	Insufficient Input Validation	Installed a malware	Remove from network (ST) Patch System (LT)	All types of impact	Application Local
001B	USE CASE 1	Misconfiguration	Installed a malware	Remove from network (ST) Patch System (MT)	All types of impact	Application Local
001 C	USE CASE 1	Design Flaw	Installed a malware	Remove from network (ST) Patch System (LT)	All types of impact	Application Local

The formal description shows that the mitigation of such risk could be set up in different ways according to the real attack vector.

6 Conclusions

The document provided an overview of the reference cyber-analysis and detection layer for the CockpitCI project.

Considering the goals of CockpitCI, different cyber security methodologies, techniques and tools were reviewed, in order to provide a broader perspective of the state-of-the-art in security technologies. This review was essential, not only to understand the best practices in the field, but also to introduce concepts that are used in the reference architecture hereby described.

The CockpitCI reference cyber analysis and detection layer hereby described was designed in order to provide the key elements specified in the project proposal, with special relevance to domain-oriented detection mechanisms and support for the Smart RTU autonomous reaction layer.

Therefore, the architecture hereby presented was designed to be as flexible as possible, in order to cope with a wide range of approaches, tools and techniques needed for effective implementation of the analysis and detection mechanisms to be incorporated into the cyber-analysis and detection layer.

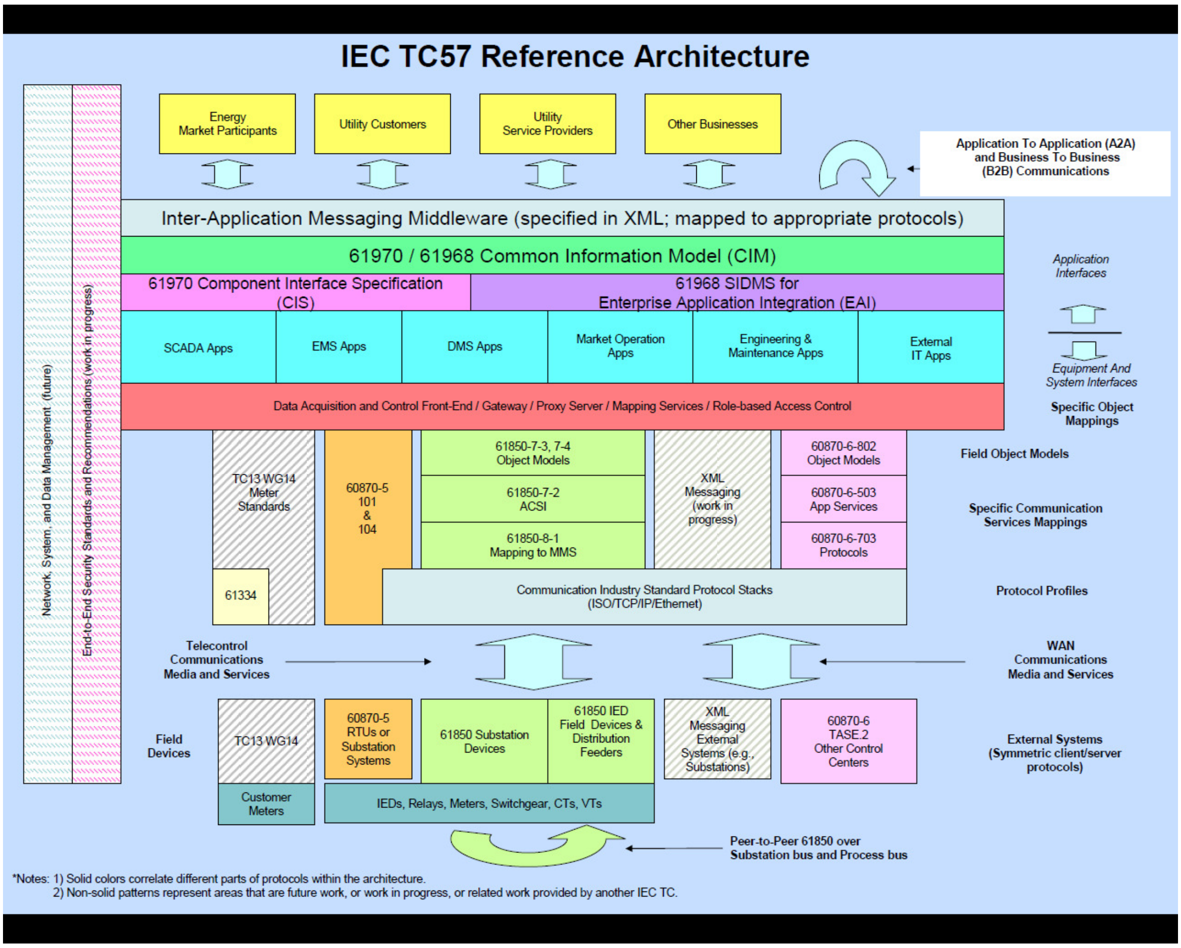
This deliverable has therefore set the stage and guidelines for the development of the technologies, interfaces and mechanisms that are going to be incorporated into the architecture and which are further specified in detail in deliverables D3.2, D3.3 and D3.4, all developed within WP3000.

7 Appendix: SCADA in the Energy Industry

SCADA/EMS (Energy Management Systems) systems are responsible for controlling, supervising, optimizing and managing energy generation and transmission. SCADA/DMS (Distribution Management Systems) perform the same functions but for power distribution networks. Both systems enable utilities to collect, store and analyze data from thousands of data points in national or regional networks, perform network modeling, simulate power operation, show faults, preempt outages, and participate in energy trading markets. Such systems are a vital part of today's power networks, enabling to handle large quantities of renewable power sources from both large and small-scale generators and to maintain grid stability. EMS and DMS represent structures based on digital equipment used by energy dispatchers to assist them in the operation control of the energy complex systems.

7.1 IEC standards

IEC's Technical Committee 57 (TC57) is responsible for developing and maintaining a set of SCADA protocol standards [IEC62351-1] for control equipment and related systems, including EMS, DMS, teleprotection and associated information exchange for real time and non-real-time information, for planning, operation and maintenance (see Figure 7-1).


IEC TC57 Reference Architecture
Energy Market Participants
Utility Customers
Utility Service Providers
Other Businesses
Application To Application (A2A) and Business To Business (B2B) Communications
Inter-Application Messaging Middleware (specified in XML; mapped to appropriate protocols)
61970 / 61968 Common Information Model (CIM)
61970 Component Interface Specification (CIS)
61968 SIDMS for Enterprise Application Integration (EAI)
SCADA Apps
EMS Apps
DMS Apps
Market Operation Apps
Engineering & Maintenance Apps
External IT Apps
Application Interfaces
Equipment And System Interfaces
Data Acquisition and Control Front-End / Gateway / Proxy Server / Mapping Services / Role-based Access Control
Specific Object Mappings
Network, System, and Data Management (future)
End-to-End Security Standards and Recommendations (work in progress)
TC13 WG14 Meter Standards
61334
60870-5 101 & 104
61850-7-3, 7-4 Object Models
61850-7-2 ACSI
61850-8-1 Mapping to MMS
XML Messaging (work in progress)
60870-6-802 Object Models
60870-6-503 App Services
60870-6-703 Protocols
Field Object Models
Specific Communication Services Mappings
Communication Industry Standard Protocol Stacks (ISO/TCP/IP/Ethernet)
Protocol Profiles
Telecontrol Communications Media and Services
WAN Communications Media and Services
Field Devices
TC13 WG14
Customer Meters
60870-5 RTUs or Substation Systems
61850 Substation Devices
61850 IED Field Devices & Distribution Feeders
IEDs, Relays, Meters, Switchgear, CTs, VTs
XML Messaging External Systems (e.g., Substations)
60870-6 TASE.2 Other Control Centers
External Systems (Symmetric client/server protocols)
Peer-to-Peer 61850 over Substation bus and Process bus

*Notes: 1) Solid colors correlate different parts of protocols within the architecture.
2) Non-solid patterns represent areas that are future work, or work in progress, or related work provided by another IEC TC.

Figure 7-1: IEC TC57 Reference Architecture (from [IEC62351-1])

Within IEC TC57, Working Group 15 (WG15) was formed to undertake the development of cyber security standards for power system communications. Its scope and purpose are to: “Undertake the development of standards for security of the communication protocols defined by the IEC TC 57, specifically the IEC 60870-5 series, the IEC 60870-6 series, the IEC 61850 series, the IEC 61970 series, and the IEC 61968 series.

The IEC 62351 standards (some under development or update) consist of:

- IEC/TS 62351-1: Introduction
- IEC/TS 62351-2: Glossary
- IEC/TS 62351-3: Security for profiles including TCP/IP
- IEC/TS 62351-4: Security for profiles including MMS
- IEC/TS 62351-5: Security for IEC 60870-5 and derivatives
- IEC/TS 62351-6: Security for IEC 61850 profiles
- IEC/TS 62351-7: Objects for Network Management
- IEC/TS 62351-8: Role-Based Access Control
- IEC/TS 62351-9: Key Management

- IEC/TS 62351-10: Security Architecture
- IEC/TS 62351-11: Security for XML Files

There is not a one-to-one correlation between the IEC TC57 communication standards and the IEC 62351 security standards. This is because many of the communication standards rely on the same underlying standards at different layers. The interrelationships between the IEC TC57 standards and the IEC 62351 security standards are illustrated in Figure 7-2.

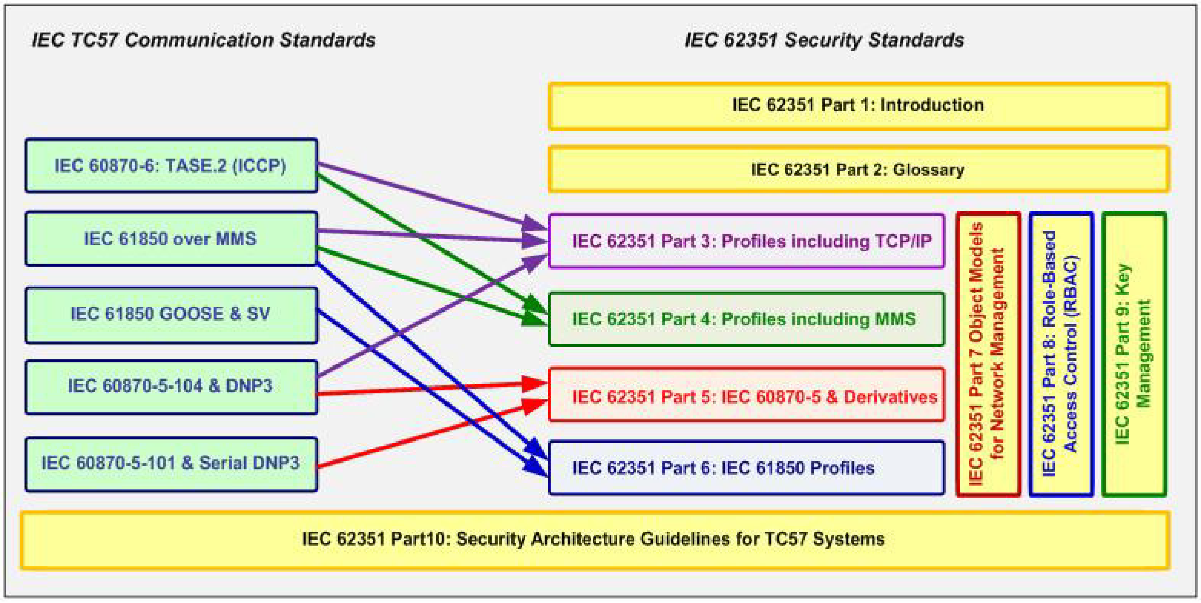

IEC TC57 Communication Standards
IEC 62351 Security Standards
IEC 62351 Part 1: Introduction
IEC 62351 Part 2: Glossary
IEC 60870-6: TASE.2 (ICCP)
IEC 61850 over MMS
IEC 61850 GOOSE & SV
IEC 60870-5-104 & DNP3
IEC 60870-5-101 & Serial DNP3
IEC 62351 Part 3: Profiles including TCP/IP
IEC 62351 Part 4: Profiles including MMS
IEC 62351 Part 5: IEC 60870-5 & Derivatives
IEC 62351 Part 6: IEC 61850 Profiles
IEC 62351 Part 7 Object Models for Network Management
IEC 62351 Part 8: Role-Based Access Control (RBAC)
IEC 62351 Part 9: Key Management
IEC 62351 Part10: Security Architecture Guidelines for TC57 Systems

Figure 7-2: Interrelationships between the IEC TC57 Standards and the IEC 62351 Security Standards (from [IEC62351-1])

IEC TC57 has developed three widely accepted protocols, and has been the source of a fourth. These protocols are:

- IEC 60870-5, which is widely used in Europe and other non-US countries for SCADA system to RTU data communications. It is used both in serial links (Part 101) and over networks (Part 104).
- DNP 3.0 which was derived from IEC 60870-5 and is in use in the US and now is widely used in many other countries as well, primarily for SCADA system to RTU data communications
- IEC 60870-6 (also known as TASE.2 or ICCP), which is used internationally for communications between control, centers and often for communications between SCADA systems and other engineering systems within control centers.
- IEC 61850 which is used for protective relaying, substation automation, distribution automation, power quality, distributed energy resources, substation to control center, and other power industry operational functions. It includes profiles to meet the ultra fast response times of protective relaying and for the sampling of measured values,

as well as profiles focused on the monitoring and control of substation and field equipment.

Together, these international standards account for close to 90% of the data communications protocols in newly implemented and upgraded power industry SCADA systems and substation automation (Modbus, Fieldbus, and other proprietary protocols are still used in older systems and in other industries).

8 References

[Aalto2008] Michael Aalto, Reuters. Available at: http://www.reuters.com/article/2008/07/30/idUS158811+30-Jul-2008+PRN20080730

[ABB2010a] ABB, "RER620 - IEC 60870-5-101/104 Communication Protocol Manual - Table 2: Supported transmission services initiated by the controlling station", 2011

[ABB2010b] ABB, "RER620 - IEC 60870-5-101/104 Communication Protocol Manual – 2.2.2 Balanced transmission", 2011

[AFTER] AFTER Project, Available at: http://www.after-project.eu/.

[AGA12] American Gas Association, AGA Report no.12 - Cryptographic Protection of SCADA Communications Part 1: Background, Policies and Test Plan (AGA12, Part 1), 2006

[Alfaro2006] J. Alfaro, I. Barrera-Caparròs, "Intercambio distribuido de alertas para la gestión de ataques coordinados", IX Reunión Española sobre Criptología y Seguridad de la Información , Barcelona, Spain, September 2006

[Allman2006] M. Allman, E. Blanton, V. Paxson, S. Shenker, "Fighting Coordinated Attackers with Cross-Organizational Information Sharing", in Proc.of Hotnets V (5th ACM Workshop on Hot Topics in Networks), Irvine, CA, November 2006.

[Allman2008] M. Allman, C. Kreibich, V. Paxson et al., "Principles for Developing Comprehensive Network Visibility", in Proc. of HotSec 2008 (3rd USENIX Workshop on Hot Topics in Security), San Jose, USA, July 2008.

[Antoniadis2002] D. Antoniadis, "LOBSTER: A European Platform for Passive Network Traffic Monitoring", in Proc. of TRIDENTCOM 2008 (4th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities), Innsbruck, Austria, March 2008

[Arpwatch] Lawrence Berkeley National Laboratory, Arpwatch tool. Available at: http://ee.lbl.gov/

[Baecher2006] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, "The nepenthes platform: an efficient approach to collect malware," in Recent Advances in Intrusion Detection, vol. 4219 of Lecture Notes in Computer Science, pp. 165–184, Springer, Berlin, Germany, 2006.

[Bailey2003] David Bailey and Edwin Wright, Practical SCADA for Industry. Great Britain: IDC Technologies, 2003.

[Bailey2005] M. Bailey, E. Cooke et al., " Data Reduction for the Scalable Automated Analysis of Distributed Darknet Traffic" in Proc. of IMC'05 (ACM SIGCOMM Internet Measurement Conference 2005), New Orleans, USA, October 2005.

[Barford2002] Barford P, Kline J, Plonka D, Ron A. A signal analysis of network traffic anomalies. Proceedings of the 2nd Internet Measurement Workshop 2002, IMW'02, 2002.

[Björkman2010] Gunnar Björkman, "The VIKING Project - Towards more Secure SCADA Systems", 2010.

[Boeing2013] The Boeing Company, "NodeBrain - An open source agent for event monitoring applications," 2013.

[Bro] Vern Paxson, Bro: A System for Detecting Network Intruders in Realtime. Computer Network Journal 23-24 (December 1999), 2435-2463.

[Brutlag2000] Brutlag J. Aberrant behavior detection in time series for network monitoring. Proceedings of the 14th USENIX conference on System administration, LISA 2000, 2000.

[Byres2005] Byres, E.; Chauvin, B.; Karsch, J.; Hoffman, D.; Kube, N.; , "The special needs of SCADA/PCN firewalls: architectures and test results," Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on , vol.2, no., pp.8 pp.-884, 19-22 Sept. 2005 doi: 10.1109/ETFA.2005.1612765

[CCECD2013] SANS Institute, "Critical Control for Effective Cyber Defense", Version 4.1, March 2013

[CERT2006] Vulnerability Note VU#190617 LiveData ICCP Server heap buffer overflow vulnerability, 2006, http://www.kb.cert.org/vuls/id/190617

[Chee-Wooi2007] Chee-Wooi Ten and Chen-Ching Liu. Cybersecurity for elettric power control and automation systems. IEEE, 2007.

[Chiesa2009] R. Chiesa, A. L.R. Pennasilico, F. Guasconi, and E. M. Tieghi. Tutto quello che avreste voluto sapere sulla protezione di reti e sistemi di controllo ed automazione...ma non avete mai osato chiedere. pdf, 2009.

[Chifflier2008] P. Chifflier, and S. Tricaud, "Intrusion Detection Systems Correlation: a Weapon of Mass Investigation", in Proc. of CanSecWest 2008 (CanSecWest Applied Security Conference 2008), Vancouver, Canada, March 2008, Available at: http://www.prelude-ids.com/fileadmin/templates/pdf/correlation-womi-cansec2008.pdf

[CIDF] Common Intrusion Detection Framework. Available at: http://gost.isi.edu/cidf/

[Clarke2004] Gordon Clarke, Deon Reynders, and Edwin Wrigth, Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems. Great Britain: IDC Technologies, 2004

[CNN2007] http://edition.cnn.com/2007/US/09/26/power.at.risk/index.html. For the video http://www.youtube.com/watch?v=fJyWngDco3g.

[CockpitCI2013] CockpitCI, Deliverable D3.2 "Research of real-time intrusion detection strategies"

[CockpitCI2013a] CockpitCI, Deliverable D3.3 "Design of the Detection Agents and Field Adaptors"

[CockpitCI2013b] CockpitCI, Deliverable D3.4 "Design of the Dynamic Perimeter"

[Cooke2005] E. Cooke, F. Jahanian, D. McPherson, "The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets", in Proc. of SRUTI 2005 (1st USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet), Cambridge, USA July 2005.

[CoreImpact] Core Security, Core Impact. Available at: http://www.coresecurity.com

[Creery2005] Creery, A.; Byres, E.J.;, "Industrial cybersecurity for power system and SCADA networks," Petroleum and Chemical Industry Conference, 2005. Industry Applications Society 52nd Annual, vol., no., pp. 303- 309, 12-14 Sept. 2005 doi: 10.1109/PCICON.2005.1524567

[CRISALIS] CRISAILS Project, http://www.ctit.utwente.nl/research/projects/international /fp7-streps/crisalis.doc/

[CRS2008] CRS Report RL32114, Botnets, Cybercrime, and Cyber terrorism: Vulnerabilities and Policy Issues for Congress, Jan. 2008.

[Cuppens2001] F. Cuppens, "Managing Alerts in a Multi-Intrusion Detection Environment", in Proc. of ACSAC 2001 (17th Annual Computer Security Applications Conference) New-Orleans, USA, December 2001.

[Cuppens2002] F. Cuppens, "Correlation in an intrusion detection process", in Proc. of SECI'02 (1st INRIA's Securité des Communications sur Internet Workshop), Tunis, Tunisia, September 2002.

[Cygwin] Cygwin project, available at: http://www.cygwin.com

[Damiani2004] E. Damiani, S. Vimercati, P. Samarati, "P2P-Based Collaborative Spam Detection and Filtering", in Proc. of P2P 2004 (4th IEEE Conference on Peer-to-Peer Computing), Zurich, Switzerland, August 2004.

[Davis2006] Davis, C.M.; Tate, J.E.; Okhravi, H.; Grier, C.; Overbye, T.J.; Nicol, D.; , "SCADA Cyber Security Testbed Development," Power Symposium, 2006. NAPS 2006. 38th North American, vol., no., pp.483-488, 17-19 Sept. 2006 doi: 10.1109/NAPS.2006.359615

[DCC] P. Vixie, "Distributed Checksum Clearinghouse", available at: http://www.rhyolite.com/anti-spam/dcc/

[Debar2007] Debar H, et al. The Intrusion Detection Message Exchange Format (IDMEF). IETF RFC 4765, 2007.

[Dewaele2007] Dewaele G, Fukuda K, Borgnat P, et al. Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures. Proceedings ACM SIGCOMM 2007 Workshop on Large-Scale Attack Defense, LSAD'07, 2007.

[DigitalBond] Digital Bond, Quickdraw SCADA IDS signatures. Available at: http://www.digitalbond.com /tools/quickdraw

[DOD2009] Aurora Overview, Provided Coutersy of DOD, November 2009, Unclassified for public Distribution.

[Douglieris2004] C. Douglieris, A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art", Published on Computer Networks, Vol. 44, Elsevier, 2004

[ESCoRTS] ESCoRTS - Security of Control and Real Time Systems. Available at: http://www.escortsproject.eu/

[ESCoRTS2008] ESCoRTS, "Survey of Stakeholders Needs - Deliverable 1.1", Dez. 2008.

[ESCoRTS2010] ESCoRTS, "R&D and standardization Road Map - Devilerable 3.2", Dez. 2010

[ESCoRTS2010a] ESCoRTS, "Survey of Existing Methods, Procedures and Guidelines - Deliverable 2.1", Jan. 2010

[ESCoRTS2010b] ESCoRTS, "Security metrics for cyber security assessment and testing - Deliverable 4.2", Ago. 2010.

[ESCoRTS2010c] ESCoRTS, "TAXONOMY of SECURITY SOLUTIONS for the SCADA Sector - Deliverable 2.2, Version 1.1", Mar, 2010.

[ESCoRTS2011] ESCoRTS, "Requirements for future cyber security laboratories - Deliverable 4.3", Jan. 2011.

[Feamster2010] N. Feamster, "Outsourcing Home Network Security", in Proc. of HomeNets'10 (2010 ACM SIGCOMM workshop on Home networks), New Delhi, India, 2010

[GAO2005] "Critical infrastructure protection report," Critical Infrastructure Protection GAO-05-434, Department of Homeland Security Faces Challenge in Fulfilling Cybersecurity Responsibilities, May 2005. [online]. Available at: http://www.gao.gov/new.items/d05434.pdf

[Garcia-Teodoro2009] García-Teodoro P, Díaz-Verdejo J, Maciá-Fernandez G, Vázquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Published in Computers & Security, Vol. 28, No. 1-2, pp. 18-28, 2009.

[GCN2013] http://gcn.com/articles/2012/11/05/agencies-join-effort-to-promote-critical-controls-for-cybersecurity.aspx, November 5, 2012 (Accessed January 7, 2013)

[Gordon2004] C. Gordon, “Modern SCADA Protocols: DNP3, 60870.5 and Related Systems - ”, 2004

[Grimes2005] M. Grimes, “SCADA Exposed,” ToorCon 7, 2005.

[Gu2008] Gu G, Perdisci R, Zhang J, Lee W. BotMiner: Clustering analysis of network traffic for protocol-and structure-independent botnet detection. Proceedings of the 17th USENIX Security Symposium, 2008.

[Hecker2009] Hecker A. and Riguidel M. (2009) ‘On the Operational Security Assurance Evaluation of Networked IT Systems’ Proceedings of the 9th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking and Second Conference on Smart Spaces, Lecture Notes in Computer Science, Volume 5764/2009 pp. 266-278 Springer-Verlag Berlin, Heidelberg

[Hes2009] Radek Hes, Peter Komisarczuk, Ramon Steenson, and Christian Seifert. The Capture-HPC client architecture. Technical report, Victoria University of Wellington, 2009.

[Homeland] Homeland security. Recommended practice: improving industrial control systems cybersecurity with defense-in-depth strategies.

[Homeland2010] Homeland security: Cyber Security Assessments of Industrial Control Systems, Centre for the protectionof national infrastructure, 2010.

[Honeyd] Honeyd – Virtual Honeypot, http://www.honeyd.org/

[Honeynet] Lance Spitzner et al: The Honeynet Project: Research Alliance, http://www.honeynet.org, Honeynet

[IDC2009] Practical Fieldbus, DeviceNet and Ethernet for Industry.: IDC Technologies, 2009.

[IEC60870-5-1] International Electrotechnical Commission, “Telecontrol equipment and systems. Part 5: Transmission protocols - Section One: Transmission frame formats”, 1990

[IEC60870-5-101] International Electrotechnical Commission, “Telecontrol equipment and systems - Part 5-101: Transmission protocols - Companion standard for basic telecontrol tasks”, 2003

[IEC60870-5-102] International Electrotechnical Commission, “Telecontrol equipment and systems - Part 5: Transmission protocols - Section 102: Companion standard for the transmission of integrated totals in electric power systems”, 1996

[IEC60870-5-103] International Electrotechnical Commission, “Telecontrol equipment and systems - Part 5-103: Transmission protocols - Companion standard for the informative interface of protection equipment”, 1997

[IEC60870-5-104] International Electrotechnical Commission, “Telecontrol equipment and systems - Part 5-104: Transmission protocols - Network access for IEC 60870-5-101 using standard transport profiles”, 2006

[IEC60870-5-2] International Electrotechnical Commission, “Telecontrol equipment and systems - Part 5: Transmission protocols - Section 2: Link transmission procedures”, 1992

[IEC60870-5-3] International Electrotechnical Commission, “Telecontrol equipment and systems - Part 5: Transmission protocols - Section 3: General structure of application data”, 1992

[IEC60870-5-4] International Electrotechnical Commission, “Telecontrol equipment and systems - Part 5: Transmission protocols - Section 4: Definition and coding of application information elements”, 1993

[IEC60870-5-5] International Electrotechnical Commission, “Telecontrol equipment and systems - Part 5: Transmission protocols - Section 5: Basic application functions”, 1995

[IEC62351-1] IEC, “IEC 62357: TC57 Architecture Part 1: Reference Architecture for Power System Information Exchange”, Second Edition Draft, Rev 6, October 2011.

[IEEE 802.1X] IEEE Standard for Local and Metropolitan Area Networks: Port based Network Access Control, IEEE 802.1X-2004, December 2004.

[IEEE1815-2010] IEEE, IEEE Standard for Electric Power Systems Communications - Distributed Network Protocol (DNP3) - IEEE 1815-2010, 2010

[Igure2006] Vinay M. Igure, Sean A. Laughter, Ronald D. Williams, Security issues in SCADA networks, Computers &amp; Security, Volume 25, Issue 7, October 2006, Pages 498-506, ISSN 0167-4048, 10.1016/j.cose.2006.03.001.

[Immunity] Immunity software, Inc., Canvas penetration testing framework. Available at: http://www.immunitysec.com

[INSPIRE] INSPIRE Project site, available at: http://www.inspire-strep.eu/

[Ioannidis2002] J. Ioannidis, S. Bellovin, “Implementing pushback: Router-based defense against DDoS attacks”, in Proc. of NDSS 2002 (Network and Distributed System Security Symposium 2002), San Diego, California, February 2002.

[ISA27005] SANS Technology Institute, “A Standard for Risk Management – ISO 27005”, Available at: https://isc.sans.edu/diary.html?date=2012-10-17

[ISA-99.00.01] American National Standard , ANSI/ISA-99.00.01-2007 - Security for Industrial Automation and Control Systems - Part 1: Terminology, Concepts, and Models, 29 Oct. 2007

[ISA99] ISA99 Commitee, “ISA-62443 Standard Series”, Available at: http://isa99.isa.org /ISA99 %20Wiki/WP_Overview.aspx

[ISO15048] ISO/IEC, "ISO/IEC Standard 15408", Available at: http://www.enisa.europa.eu /activities/risk-management/current-risk/laws-regulation/rm-ra-standards/iso-iec-standard-15408

[ISO15443] ISO, "ISO 15443 Standard", Available at: http://www.iso.org/iso/cata logue_detail.htm?csnumber=39271

[ISO18043] ISO, "ISO 18043 Standard", Available at: http://www.iso.org/iso/catalo gue_detail?csnumber=35394

[ISO19791] ISO, "ISO 19791 Standard", Available at: http://www.iso.org/iso /home/store/catalogue _tc/catalogue_detail.htm?csnumber=52905

[ISO27001] ISO, "ISO 27001 Standard", Available at: http://www.27000.org/iso-27001.htm

[Jansen2009] Jansen W., 2009. Directions in Security Metrics Research. National Institute of Standards and Technology, Special publication #NISTIR 7564, NIST, Gaithersburg, MD.

[Jay2003a] M. Jay, “Comparison of protocols used in remote monitoring: DNP 3.0, IEC 870-5-101 & Modbus - Figure 6 - ASDU frame details for IEC 870-5-101”, 2003

[Jay2003b] M. Jay, “Comparison of protocols used in remote monitoring: DNP 3.0, IEC 870-5-101 & Modbus - Table 1 - Comparison of DNP 3.0, IEC 870-5-101 and Modbus”, 2003

[Kahn1998] Kahn C, Porras P, Staniford-Chen S, Tung B. A Common Intrusion Detection Framework, 1998.

[Kang2011] D.J. Kang, J.J. Lee, B.H. Kim, D. Hur, Proposal strategies of key management for data encryption in SCADA network of electric power systems, International Journal of Electrical Power &amp; Energy Systems, Volume 33, Issue 9, November 2011, Pages 1521-1526, ISSN 0142-0615, 10.1016/j.ijepes.2009.03.004.

[Katti2005] S. Katti, B. Krishnamurthy, D. Katabi, “Collaborating Against Common Enemies”, in Proc. of IMC‘05 (ACM SIGCOMM Internet Measurement Conference), New Orleans, USA, October 2005.

[Kong2006] J. Kong et al., “Scalable and Reliabile Collaborative Spam Filters: Harnessing the Global Social Email Networks”, in Proc. of WWW 2006 (15th International World Wide Web Conference) Workshop on the Weblogging Ecosystem, Edinburgh, UK, May 2006.

[Koutepas2004] G. Koutepas, F. Stamatelopoulos, B. ,Maglaris , “Distributed Management Architecture for Cooperative Detection and Reaction to DDoS Attacks”, published in JNSM (Journal of Network and Systems Management), Vol. 12, No. 1, March 2004

[Krishnamurthy2003] Krishnamurthy B, Sen C, Zhang Y, Chen Y. Sketchbased change detection: methods, evaluation, and applications. Proceedings of the Internet Measurement Conference 2003, IMC'03, 2003.

[Krutz2006] Ronald L. Krutz, Securing Scada Systems. USA: Wiley Publishing, Inc., 2006.

[Lakhina2004] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies" in Proc. of ACM SIGCOMM '04, Portland, USA, August 2004

[Lakhina2004] Lakhina C, Crovella M, Diot C. Diagnosing network-wide traffic anomalies. Proceedings of ACM SIGCOMM'04, 2004.

[Lakhina2004a] A. Lakhina, K. Papagiannaki, M. Crovella, et al., "Structural analysis of network traffic flows", in Proc. of SIGMETRICS'2004 (ACM Conference on Measurement and Modeling of Computer Systems 2004), New York, USA, June 2004.

[Li 2011] Li Li; Sun, Hua; Zhenyu Zhang, "The research and design of honeypot system applied in the LAN security," Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on , vol., no., pp.360,363, 15-17 July 2011

[Lian2011a] MAYR Software, "Types of control field formats - I-Format", 2011

[Lian2011b] MAYR Software, "Types of control field formats - S-Format", 2011

[Lian2011c] MAYR Software, "Types of control field formats - U-Format", 2011

[Libpcap] Libpcap, TCPDUMP/LIBPCAP public repository, [online] http://www.tcpdump.org/

[Marinova-Boncheva2007] Marinova-Boncheva V. A Short Survey of Intrusion Detection Systems. Published in Problems of Engineering, Cybernetics and Robotics, Vol. 58, Institute of Information Technologies – Bulgarian Academy of Sciences, 2007.

[Mazel2011] Mazel J, Casas P, Owezarski P. Sub-space clustering & evidence accumulation for unsupervised network anomaly detection. Proceeding of the 3rd COST TMA International Workshop on Traffic Monitoring and Analysis, TMA'11, 2011.

[Mazel2011a] Mazel J, Casas P, et al. Sub-Space Clustering, Inter-Clustering Results Association & Anomaly Correlation for Unsupervised Network Anomaly Detection. Proceedings the 7th International Conference on Network and Service Management, CNSM 2011, 2011.

[McAfee2011] McAfee Foundation professional service and McAfee Labs. Global energy cyberattack: "night dragon", 2011

[Metasploit] Metasploit penetration testing. Available at: http://www.metasploit.com/

[MICIE] MICIE project, Available at: http://www.micie.eu

[Minich] Christofer Minich and Howard Ragunton. Surviving a cyber attack on your SCADA system.

[Modbus] Modbus Organization, Disponível em: http://www.modbus.org/ (Último acesso: Dez. 2011)

[Modbus2006] Modbus-IDA, "Modbus Application Protocol Specification V1.1b", Dez. 2006

[Modicon1996] MODICON, Inc., Industrial Automation Systems, “Modbus Protocol Reference Guide”, 1996

[Nahorney2003] Benjamin Nahorney, Symantec. Available at: http://www.symantec.com/security_response/writeup.jsp?docid=2003-081909-2118-99

[Nai2007] I. Nai Fovino, M. Masera and A. Decian, Integrating Cyber Attack within Fault Trees, In Proceeding of the European Safety and Reliability Conference (ESREL), June 25–27, 2007, Stavanger.

[NERC] “Cyber security standards,” NERC. [online]. Available at: http://www.nerc.com /~filez/standards/Cyber-Security-P ermanent.html, 2006.

[Nessus] Nessus vulnerability scanner. Available at: http://www.tenable.com/products /nessus

[Niland2003] Marty Niland, In Computer Virus Brings Down Train Signals.Information Week. Available at: http://www.informationweek.com/news/

[NISCC2005] NISCC, NISCC good practice guide on firewall deployment for SCADA and process control networks.: National Infrastructure Security Cordinaton Centre (NISCC), February 2005.

[NIST] NIST. Guide to industrial control system (ics) security.

[NIST2005]NIST. Keith Stouffer, Joe Falco, Karen Scarfone, “Guide to Industrial Control System (ICS) security” (Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC)), NIST Special Publication 800-82, June 2011.

[NIST2011]NIST. Peter Mell, Karen Kent, and Joseph Nusbaum, “Guide to Malware Incident Prevention and Handling”, NIST Special Publication 800-83, November 2005.

[O’Murchu2011] L. O’Murchu N. Falliere. W32.Stuxnet dossier, Symantec White Paper, February 2011.

[OSSEC] OSSEC HIDS. Available at: http://www.ossec.net

[Ouedraogo2011] Ouedraogo, M., Mouratidis, H., Hecker A., Bonhomme C., Khadraoui, D., Dubois E., Preston D. (2011) ‘A new approach to evaluating security assurance’, Proceedings of the international conference on Information Assurance and Security (IAS), pp. 215-221.

[Ouedraogo2012] Ouedraogo M., Khadraoui, D., Mouratidis, H., Dubois, E. (2012) 'Appraisal and reporting of security assurance at operational systems level' Journal of software and system and Software85(1), pp. 193-208, Elsevier.

[Ouedraogo2013] Ouedraogo, M., Savola, R., Mouratidis, H., Preston D, Kadraoui, D., Dubois, E. (2013)'Taxonomy of quality metrics for assessing assurance of security correctness', Software Quality Journal, Volume 21, Issue 1, pp 67-97 Springe.

[Pauli2003] Pauli, S.; Krahl, B.; Leuschner, B.; , "A guide to specifying, justifying and installing substation monitoring and control systems," Petroleum and Chemical Industry Conference, 2003. Record of Conference Papers. IEEE Industry Applications Society 50th Annual , vol., no., pp. 71- 80, 15-17 Sept. 2003 doi: 10.1109/PCICON.2003.1242600

[Perdisci2010] Perdisci R, Lee W, Feamster N. Behavioral Clustering of HTTP-Based Malware. Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation, NSDI'10, 2010.

[Pollet2002] Pollet, J. (August 8, 2002). SCADA Security Stratege, PlantData Technologies, http://www.plantdata.com/ SCADA/Security/Strategy.pdf.

[Poulsen2003] Kevin Poulsen, In Slammer worm crashed Ohio nuke plant network. Security Focus. Available at: http://www.securityfocus.com/news/6767

[PRECYSE] PRECYSE project, Available at: http://cordis.europa.eu/projects/rcn/61016_en.html.

[Profibus] PROFIBUS & PROFINET International, Process Field Bus Standard. Available at http://www.profibus.com/

[Provos2004] Provos, Niels. "A Virtual Honeypot Framework." In Proceedings of the 13th USENIX Security Symposium. 2004. 1-14.

[Pvbrowser] Pvbrowser Project site, , available at: http://pvbrowser.de/

[Pyzor] Pyzor. http://pyzor.sourceforge.net

[RaspberryPi] Raspberry Pi - An ARM GNU/Linux box. Available at http://www.raspberrypi. org/

[Razor] V. Prakash ,"Vipul's Razor", available at: http://razor.sourceforge.net

[Riden2010] Riden, J., Seifert, C., "A Guide to Different Kinds of Honeypots", Symantec Connect, November 2010, available at: http://www.symantec.com/connect/articles/guide-different-kinds-honeypots

[Roberts2005] Roberts, P. “Zotob, PnP Worms Slam 13 DaimlerChrysler Plants”, eweek.com, August 18, 2005, http://www.eweek.com/c/a/Security/Zotob-PnP-Worms-Slam-13- DaimlerChrysler-Plants/

[Ryu2007] Ryu, D., Kim, H., Shin, S., & Nahm, S. (2007). ‘‘Security Vulnerabilities of Critical Infrastructure Systems,’’ World Conference on Safety of Oil and Gas Industry (WCOGI), Gyeongju, Korea, Vol. 2, pp. 218–222.

[Sal2009] Doug Salmon, Mark Zeller, Armando Guzmán, Venkat Mynam, and Marco Donolo, “Mitigating the Aurora Vulnerability With Existing Technology”, Schweitzer Engineering Laboratories, Inc, 2009.

[SANS] SANS, “Intrusion Detection FAQ: What is the Role of Security Event Correlation in Intrusion Detection?”, Available at: http://www.sans.org/security-resources/idfaq/role.php

[SANS2013], Matthew E. Luallen, SANS SCADA and Process Control Security Survey, SANS White Paper, SANS Analyst Program, February 2013.

[Saw2013] Shaw T. William (2013) Scada systems vulnerabilities to cyber attack, available at: http://www.electricenergyonline.com/?page=show_article&article=181

[SCADAHacker] SCADAHacker, SCADA Modules for Metasploit. Available at: http://scadahacker.com/resources/msf-scada.html

[Shelia] Shelia client honeypot, Available at: http://www.cs.vu.nl/~herbertb/misc/shelia/

[Silveira2010] Silveira F, Diot C. Urca: Pulling out anomalies by their root causes. Proceedings of the 29th IEEE Conference on Computer Communications, INFOCOM’10, 2010.

[Simmons2009] Chris Simmons, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, Qishi Wu, “AVOIDIT: A Cyber Attack Taxonomy”, Department of Computer Science, University of Memphis, Memphis, TN, USA, August 2009.

[Slay2007] Slay, J., and Miller, M. Lessons learned from the maroochy water breach. In Critical Infrastructure Protection (November 2007), vol. 253/2007, Springer Boston, pp. 73–82.

[Snort] Snort IDS. Available at: http://www.snort.org

[Soule2005] Soule A, Salamatian K, Taft N. Combining filtering and statistical methods for anomaly detection. Proceedings of the Internet Measurement Conference 2005, IMC’05, 2005.

[Spafford1994] Kim G, Spafford E. The Design and Implementation of Tripwire: A File System Integrity Checker. Proceedings of the 2nd ACM Conference on Computer and Communications Security, SIGCOMM’94, 1994.

[Spitzner2002] Spitzner, Lance. Honeypots: Tracking Hackers. Addison-Wesley Professional, 2002.

[Staniford2002] Staniford S, Hoagland J, McAlerney J. Practical automated detection of stealthy portscans. Published on the Journal of Computer Security (JCS), IOS Press, Vol. 10, 2002.

[Staniford-Chen1998] Staniford-Chen S, Tung B, Schnackenberg, D. The Common Intrusion Detection Framework (CIDF). Proceedings of the 1998 Information Survivability Workshop, ISW'98, 1998.

[Suh2005] K. Suh, Y. Guo, et al. "Locating network monitors: complexity, heuristics, and coverage", in Proc. of INFOCOM'2005 (the 24th IEEE Conference on Computer Communications),Miami, USA,March 2005

[Symantec2011] W32.Duqu. The precursor to the next Stuxnet, Symantec White Paper, November 2011.

[Ten2008] Chee-Wooi Ten; Chen-Ching Liu; Manimaran, G.;, "Vulnerability Assessment of Cybersecurity for SCADA Systems," Power Systems, IEEE Transactions on , vol.23, no.4, pp.1836-1846, Nov. 2008 doi: 10.1109/TPWRS.2008.2002298

[TiangleMicroworks1999] Andrew C. West B.E., "Communication Standards in Power Control", 1999

[Torkilseng2006] A. Torkilseng and G. Ericsson, "Some guidelines for developing a framework for managing cybersecurity for an electric power utility," ELECTRA Report, Oct. 2006.

[Triangle2002] Triangle MicroWorks, Inc, DNP3 Overview, Raleigh, North Carolina, 2002, http://www.trianglemicroworks.com/documents/DNP3_Overview.pdf

[Tung2001] Tung B, et al. The Common Intrusion Detection Framework Specification, 2001.

[Vaarandi2013] Risto Vaarandi, "Sec - simple event correlator," 2013.

[Vandoorselaere2008] Vandoorselaere Y. Prelude Universal SIM: State of the Art. Presented at the Libre Software Meeting 2008, 2008. Available at: www.preludetechnologies. com/fileadmin/templates/pdf/RMLL2008.pdf

[Verba2008] Verba, J.; Milvich, M.;, "Idaho National Laboratory Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS)," Technologies for Homeland Security, 2008 IEEE Conference on , vol., no., pp.469-473, 12-13 May 2008

[Verba2008] Verba, J.; Milvich, M.;, "Idaho National Laboratory Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS)," Technologies for Homeland Security, 2008 IEEE Conference on , vol., no., pp.469-473, 12-13 May 2008 doi: 10.1109/THS.2008.4534498

[VIKING] VIKING Project. Available in: http://www.vikingproject.eu/

[VIKING2010] VIKING, "VIKING Cities Simulator (ViCiSi)". Vital Infrastructure, NetworKs, INformation and Control System ManaGement, 2010

[VIKING2010a] VIKING, "Cyber Security Modeling Language (CySeMoL)". Vital Infrastructure, NetworKs, INformation and Control Systems ManaGement, 2010

[VIKING2010b] VIKING, "VIKING Testbed Overview". 2010

[VIKING2010d] VIKING Project, "Report D2.3", 2010.

[Wan2002] K. Wan, R. Chang, "Engineering of a global defense infrastructure for DDoS attacks", in Proc. of ICON 2002 (10th IEEE International Conference on Networks), Singapore, August 2002.

[Wang2006] Y. Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S.T. King, "Automated Web Patrol with Strider HoneyMonkeys: Finding Web Sites That Exploit Browser Vulnerabilities", ;in Proc. NDSS, 2006.

[Weiqing2010a] T. Weiqing, "Realization of IEC 60870-5-104 Protocol in DTU - Fig.2 network reference model", 2010

[Weiqing2010b] T. Weiqing, "Realization of IEC 60870-5-104 Protocol in DTU - Fig. 3 APDU of the defined telecontrol companion standard - Figure 5.25 Application function codes", 2010

[Wireshark] Wireshark project. Available at: http://www.wireshark.org/

[Zhu2011] Bonnie Zhu, Anthony Joseph, and Shankar Sastry, "A taxonomy of Cyber Attacks on SCADA Systems," , Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, 2011

[Zhu2011] Bonnie Zhu, Anthony Joseph, and Shankar Sastry, "A taxonomy of Cyber Attacks on SCADA Systems," , Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, 2011