FP7-SEC-2011-1 Project 285647

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

D2.1 - Overview of modelling techniques and tools for SCADA systems under cyber attacks

Organisation name of lead contractor for this deliverable
ENEA

General information	
Submission date	3 July 2012
Dissemination level	Public
State	Final version
Work package	WP2000 - Modelling and prediction of QoS of interdependent SCADA and Telco Networks facing cyber attacks
Task	Task 2001 - Overview of modelling techniques and tools to represent SCADA systems under cyber attacks
Delivery date	30 June 2012

Editors

Name	Organisation
E. Ciancamerla, M. Minichino	ENEA

Authors

Name	Organisation
E. Ciancamerla, A. Di Pietro, M. Minichino, S. Palmieri	ENEA
M. Ouedraogo	Henry Tudor
S. Iassinovski	Multitel
T. Cruz, E. Monteiro, J. Proença, P. Simões	University of Coimbra
C. Foglietta, S. Panzieri	Roma 3

Reviewers

Name	Organisation	Date
S. Iassinovski	Multitel	28-06-2012
M. Aubigny	iTrust	01-07-2012

Executive Summary

This document provides an overview of modelling techniques and tools able to represent Industrial Control Systems (ICS) under cyber attacks.

ICS include supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC). They are typically used in industries such as electric, water and wastewater, oil and natural gas, etc. SCADA systems are generally used to control dispersed assets using centralized data acquisition and supervisory control. DCS are generally used to control production systems within a local area such as a factory using supervisory and regulatory control. PLC are generally used for discrete control for specific applications and generally provide regulatory control.

These control systems are vital to the operation of critical infrastructures that are often highly interconnected and mutually dependent systems.

Before going into details on the overview of modelling techniques and tools to represent ICS and SCADA and their behaviour under cyber attacks, this document intends also to give details on a minimum, preliminary, but still huge, context. That is needed to make the document auto consistent and to help even a not expert reader in understanding the main issues of modelling techniques and tools able to represent Industrial Control Systems (ICS) under cyber attacks.

Particularly, the document deals with the following topics.

Glossary. It is an extended glossary extracted by sector standards and guidelines.

CockpitCI vision. CockpitCI system will be feed by prediction models of QoS delivered to CI customers in nominal conditions and under cyber attacks of SCADA and enterprise network. Within the project, modelling techniques able to represent cyber attacks, their exploitation throughout cyber vulnerabilities of Critical Infrastructures, up to penetration within Industrial Control Systems and SCADA will be investigated. According to the project aim a special attention has been paid to the ability of such techniques and tools to predict the impact of successful attacks on the Industrial Control Systems of which SCADA systems is a subset, and in turn on the Quality of Service delivered by the target CI which is either functionally or cyber interdependent with other CIs.

Malware classification. Malware has become the most significant external threat to most systems, causing widespread damage and disruption, and necessitating extensive recovery efforts within most organizations. Several major forms of malware, including viruses, worms, Trojan horses, malicious mobile code, blended attacks, spyware tracking cookies, and attacker tools such as backdoors and rootkits have been addressed.

Stealing information and starting an attack. The techniques used for steal information and start a cyber attack to a SCADA system are not too different to an ICT system. Typical attack phases, such as : Password guessing, Port scanning, Exploitation, Man-In-The-Middle (MITM), Denial of Service (DoS) have been considered.

Industrial Control System within a CI. A hierarchy of logical level characterize the ICS within a Critical Infrastructure. Differences and similarities in cyber security between ICS

and ICT (Information and Communication Technology) systems are highlighted. Initially, ICS had little resemblance to traditional 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.

ICS cyber threats, vulnerabilities and attacks. Originally, ICS implementations were susceptible primarily to local threats because many of their components were in physically secured areas and the components were not connected to IT networks or systems. However, the trend toward integrating ICS systems with IT networks provides significantly less isolation for ICS from the outside world than predecessor systems, creating a greater need to secure these systems from remote, external threats.

ICS security policies and solutions. Major security objectives for an ICS implementation should include the following: a) restricting logical access to and protecting the ICS network and network activity. This includes using a demilitarized zone (DMZ) network architecture with firewalls to prevent network traffic from passing directly between the corporate and ICS networks, and having separate authentication mechanisms and credentials for users of the corporate and ICS networks, Intrusion Detection/Prevention Systems and Honeypots/Honeynets; b) protecting individual ICS components from exploitation. This includes deploying security patches; disabling all unused ports and services; restricting ICS user privileges to only those that are required for each person's role; tracking and monitoring audit trails; and using security controls such as antivirus software and file integrity checking software where technically feasible to prevent, deter, detect, and mitigate malware.

ICS cyber security: modelling techniques and tools. Cyber security methodologies, models and tools are fundamentally based on identification of attacker profiles, attack objectives, attack steps characterization, spreading throughout ICS network and consequences on CI customers. In this view and having in mind the main objective of CockpitCi project, different cyber security methodologies, models and tools, used as a single package to address specific aspects of the attack scenario, and/or integrated together to afford the whole attack scenario are discussed. At the state of the art, no single modelling technique has the modelling power and the analytical tractability to adequately deal with the modelling and early prediction of QoS of SCADA system facing adverse events, such as cyber attacks, and accounting cyber interdependency along CI ICT backbone. As a consequence, for analyzing ICS under cyber attacks and the related consequences on CI (i.e. Power grid) services to customers, we distinguish four kinds of models each one requiring specialized methods and tools which, in turn, could rely on specialized or not (general) modelling formalisms: 1) Attacks/attacker/vulnerability models (attack/vulnerability trees, Petri nets, Game theory); 2) ICS & enterprise network models (network simulators/emulators); 3) CI models (i.e. electrical models by power flow simulators); 4) Composite models to represent more than one aspect of the attack scenario (at least two different kinds of the previous models) till the whole attack scenario (i.e. attacks model plus ICS & enterprise network model plus CI model), which may require more than one (Hybrid versus homogeneous) method and tool. Also, several tools which cover partially or as whole the above methods and models, have been overviewed. Many of them rely on stochastic approach such as Petri nets, Game theory, Markov chains, Bayesian networks, Monte Carlo methods; other ones rely on different approaches such as Agent based simulation, discrete event simulation, etc. Different comparison paradigms could be used to compare the modelling techniques and tools resulting from this overview. Modelling formalisms could be ranked according to different criteria i.e. their modelling power against analytical tractability or by their ability to represent any part of the scenario to be represented within CockpitCI project (attacker profiles, attack objectives , ICS&ICT vulnerabilities, attack steps characterization , malware spreading throughout ICS network and consequences on quality of service delivered to CI customers). However, the meaning of this overview is solely to offer to CockpitCI modellers possible other modelling paths and tools that may enrich their own modelling formalisms and tools, covering complementary aspects, according to their feeling of the project and their expertise. A complete ranking of modelling formalisms that will be exercised on CockpitCI reference scenario will be provided at a later stage of the project within the appropriate WP2000 deliverables.

Cyber security toolkits. Many attackers use toolkits containing different types of utilities and scripts that can be used to probe and attack systems, such as packet sniffer, port scanners, vulnerability scanners, password crackers, remote login programs and attack programs and scripts. Many of such toolkits with a special attention to their plugins for SCADA systems have been investigated and discussed.

ICS security test beds. The meaning of ICS security test bed in literature is multifold. Two main aspects of ICS security test bed are dealt in this document: i) program with the mission to reduce the risk of energy distruption due to cyber attacks on ICS; ii) an hybrid framework composed by modelling tools and physical devices with the aim of understanding cyber security aspects of ICS and eventually, their impact on service delivered by Critical Infrastructure under control. In any testbed, modelling techniques and tools are largely used and integrated with physical devices. This deliverable addresses modelling techniques and tools used in testbeds.

Cyber security within EU projects. Many EU projects deal with cyber security aspects common to CockpitCi project and are partially into the scope of this deliverable. Some of them, such as AFTER, CRISALIS, PRECYSE, SAFEGUARD and VIKING are summarised in the document.

ICS cyber security standards and guidelines. The contents of this overview has been also supported by the most known standards and guidelines on ICS cyber security, as reported.

Table of contents

List of figures

List of table

1 Introduction

The document presents an overview of modeling techniques and tools able to represent SCADA systems under cyber attacks within a Critical Infrastructure (CI). It is not a general document on cyber security, but it is rather a document which is within the scope of CockpitCI vision [80].

According to such a vision, a special attention has been paid to the ability of such techniques and tools to predict the impact of successful attacks on the Industrial Control Systems (ICS), of which SCADA systems is a subset, and in turn on the Quality of Service delivered by the target CI, which is either functionally or cyber interdependent with other CIs.

Within this document, Industrial Control Systems are intended as 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 or railway transportation.

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.

Industrial Control Systems constitute a strategic asset against the rising potential for catastrophic terrorist attacks affecting Critical Infrastructures. In the last decade, these systems have been facing a notable number of incidents, including the manifestation of Stuxnet which raised a lot of concerns and discussions among all the actors involved in the field.

This overview describes into detail the state of modelling techniques and tools able to represent SCADA systems under cyber attacks within a Critical Infrastructure (CI), within the scope of CockpitCI vision [80].

Before going into details on the overview of modelling techniques and tools, this document intends also to give details on a minimum, preliminary, but still huge, context. That is needed to make the document auto consistent and to help even a not expert reader in understanding the main issues of modelling techniques and tools able to represent Industrial Control Systems (ICS) under cyber attacks.

1.1 CockpitCI vision

CockpitCI is in line with the MICIE project [36] of which it resumes the main concept, i.e. that by increasing the cooperation among infrastructures it is possible to provide the operator with a better situation awareness in the presence of adverse events and therefore increase the CI level of service (business continuity). CockpitCI proposes this concept again in a wider operational range which addresses now not only adverse events but also cyber events. The CockpitCI system will be feed by prediction models of QoS delivered to CI customers in nominal conditions and under cyberattacks of SCADA and enterprise network.

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). The CockpitCI vision is that the convergence among business continuity and cyber security is possible with positive fallouts for all the involved players. From the point of view of security staff benefits will arise thanks to the availability of new security data coming from the process network. Such data will be collected by local SCADA-oriented detection agents able to recognize traffic anomalies or intrusions attempts. Then, they will be merged, to build a wider cyber awareness, with the traditional ICT networks security related data. From the business point of view, a near real-time risk evaluation capability, exploiting also the previously built cyber awareness, will allow a clever reaction by SCADA operators to possible cyber threats and the avoidance of large domino effects. Starting from the improved risk definition, it will be also possible, for the stakeholder, to have a better tailored definition of service level agreement (and then contracts) with its customers. It is not just a question of putting together the two worlds of SCADA industrial control systems and cyber-security, but of reshaping the boundaries of each and blending the two by taking advantage of each other strengths.

Such global awareness will be fostered by fusing the information which originates from the various control rooms of the infrastructure, from the control rooms of interdependent CIs, from the control rooms at national level which are again connected with the intelligence at national and transnational level. The cyber issue is not a local problem which may be confined in a restricted boundary, but it is rather a transnational problem which goes beyond national boundaries. Therefore, the various functions of the CockpitCI tool must not be isolated.

CockpitCI will make a further step ahead by putting together the local perspective provided by information collected from the field equipment with the global perspective at CI level: the local perspective refers to the smart elements at the field level which will monitor equipment and devices and perform cyber threat detection and eventually start an automatic reaction; the global perspective refers to the wider perspective on the state of the System of Systems which, thanks to increased cooperation among infrastructures and shared interdependency models, is wider compared to previsions that can be generated by sector specific and isolated simulators. Putting the two levels to work together will be the basis for a smarter reaction capability, aiming at a graceful degradation, aiming at understanding 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 Aim of the document

The CockpitCI system will be feed by prediction models of QoS delivered to CI customers in nominal conditions and under cyber attacks of SCADA and enterprise network.

Within the project, modelling techniques able to represent cyber attacks, their exploitation throughout cyber vulnerabilities of Critical Infrastructures, up to penetration within Industrial Control Systems and SCADA will be investigated. According to the project aim a special attention has been paid to the ability of such techniques and tools to predict the impact of successful attacks on the Industrial Control Systems of which SCADA systems is a subset, and in turn on the Quality of Service delivered by the target CI which is either functionally or cyber interdependent with other CIs.

Due to the continuous evolution of cyber attacks and their technological targets, the modelling process within the CockpitCI project will be tuned by an overview of modelling, methods, techniques, software tools able to represent attackers behaviour, attack process, attack vectors, Industrial Control Systems in nominal conditions and under cyber attacks with special focus on SCADA systems and finally the Critical Infrastructure.

This overview is proceeded by a discussion of topics, the concepts and objects to be modeled/reproduced as detailed into the document structure.

The final aim of this overview is that a certain subset of modeling techniques and tools could enrich the modeling framework already available within the expertise of modeling actors of CockpitCI project.

Along the document several aspects of ICS and CI are accounted as detailed in the document structure and the overview of modelling techniques and tool has been conducted having in mind multiform facets of the reviewed modelling techniques and tools, which effectiveness is typically shown by case studies/applications :

1. Inaccurate assumptions limit the applicability of the results.

2. The selection of accurate datasets is necessary to ensure that modelling effort can be transitioned to actual test beds.

3. Cyber/Physical Network Data Sets: the development of open and accurate models of the networks and traffic are necessary to ensure that research efforts accurately represent realistic system implementations. The development of accurate network models should include realistic network topologies, communication protocols, temporal data requirements, supported power applications, and physical power system.

4. Cyber Attack Data Sets: Along with accurate network models, accurate information about possible cyber attacks is necessary to ensure that researchers are able to understand current threats and attacker techniques. Accurate attack data has main applications including the development of intrusion detection systems and intrusion tolerant architectures.

5. Realistic Testbeds. Even realistic testbeds are based on composite simulation environment and actual devices. The boundary between realistic test beds and modelling environments is not so sharp. Having accurate data sets is critical to designing more attack resilient systems, realistic testbed are also required to explore cyber-physical interdependencies and their resulting security impacts. Additionally, testbeds provide a platform where research and vendor products can be evaluated with simulated power system requirements. Realistic testbeds are also presented with a special focus on their simulated part.

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. Just public documents have been considered and they are properly referred along this document.

1.3 Document Structure

The chapters of the document respectively deal with:

- Chapter 2 deals with Malware classification. Several major forms of malware, including viruses, worms, Trojan horses, malicious mobile code, blended attacks, spyware tracking cookies, and attacker tools such as backdoors and rootkits have been addressed.
- Chapter 3 deals with the techniques used for stealing information and starting a cyber attack to a SCADA system.
- Chapter 4 deals with the specificity of ICS within Critical Infrastructures and highlights security differences and similarities between ICS and ICT systems.
- Chapter 5 deals with ICS cyber threats, vulnerabilities and attacks. Originally, ICS implementations were susceptible primarily to local threats because many of their components were in physically secured areas and the components were not connected to IT networks or systems.
- Chapter 6 deals with ICS security policies and solutions. Major security objectives for an ICS implementation should include restricting logical access to and protecting the ICS network and network activity and ICS components from exploitation too.
- Chapter 7 is the heart of the document and deals with ICS cyber security modeling techniques and tools. Cyber security methodologies, models and tools are fundamentally based on identification of attacker profiles, attack objectives , attack steps characterization , spreading throughout ICS network and consequences on CI customers.
- Chapter 8 deals with Cyber security toolkits. Many attackers use toolkits containing different types of utilities and scripts that can be used to probe and attack systems, such as packet sniffer, port scanners, vulnerability scanners, password crackers, remote login programs and attack programs and scripts. Many of such toolkits with a special attention to their plugins for SCADA systems have been investigated and discussed.
- Chapter 9 deals with ICS security test beds. ICS security test beds aim to understand cyber security aspects of ICS and eventually, their impact on service delivered by Critical Infrastructure under control.
- Chapter 10 deals with cyber security within EU projects
- At the last, in Chapter 11 an investigation of the most known standards and guidelines on ICS cyber security has been reported.

1.4 Glossary

The purpose of the glossary is to identify and define an unambiguous vocabulary of terms which will be used along this document. Particularly terms and definitions have been mainly taken from:

- Federal Information Processing Standards (FIPS) PUB 140-2, (2001) —Security Requirements for Cryptographic Modules, Section 2, Glossary of Terms and Acronyms, U.S. National Institute of Standards and Technology.

- SANS Glossary of Terms used in Security and Intrusion Detection, May 2003, http://www.sans.org/resources/glossary.php

- ISA‑62443.01.01 (99.01.01) – 22 – Draft 1, Edit 2 - December 2011

- ISO 27000:2009

To be sure to be in line with standardization work, the present document wants also take into account the terminology defined by ISO experts at last stage of editorial works, especially for the two following text at almost last stage of edition:

- FDIS ISO 27032 The text is in its final stage at ISO (stage 50.60). The final has been approved on 28.06.2012 and the text should be published in few weeks.

- FDIS ISO 29115 ITU-T Recommendation X.1254, International Standard ISO/IEC FDIS 29115, *Information technology — Security techniques — Entity authentication assurance framework*. This text is has been sent to ITTF for 2-month FDIS ballot processing in May*.*

access

ability and means to communicate with or otherwise interact with a system in order to use system resources. Access may involve physical access (authorization to be allowed physically in an area, possession of a physical key lock, PIN code, or access card or biometric attributes that allow access) or logical access (authorization to log in to a system and application, through a combination of logical and physical means)

access control

protection of system resources against unauthorized access; a process by which use of system resources is regulated according to a security policy and is permitted by only authorized entities (users, programs, processes, or other systems) according to that policy .

access control list

a list of permissions attached to an object. An access control list (ACL) specifies which users or system processes are granted access to objects, as well as what operations are allowed on given objects. Each entry in a typical ACL specifies a subject and an operation. For instance, if a file has an ACL that contains (Alice, delete), this would give Alice permission to delete the file.

asset

physical or logical object owned by or under the custodial duties of an organization, having either a perceived or actual value to the organization. In the case of industrial automation and control systems the physical assets that have the largest directly measurable value may be the equipment under control.

There are many types of assets, including: (a) information; (b) software, such as a computer program; (c) physical, such as computer; (d) services; (e) people, and their qualifications, skills, and experience; and (f) intangibles, such as reputation and image.

attack

assault on a system that derives from an intelligent threat, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system. There are different commonly recognized classes of attack:

- An "active attack" attempts to alter system resources or affect their operation.
- A "passive attack" attempts to learn or make use of information from the system but does not affect system resources.
- An "inside attack" is an attack initiated by an entity inside the security perimeter (an "insider") , i.e., an entity that is authorized to access system resources but uses them in a way not approved by those who granted the authorization.
- An "outside attack" is initiated from outside the perimeter, by an unauthorized or illegitimate user of the system (including an insider attacking from outside the security perimeter). Potential outside attackers range from amateur pranksters to organized criminals, international terrorists, and hostile governments.

attack potential

perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker’s expertise, resources and motivation

attack vector

path or means by which an attacker can gain access to a computer or network server in order to deliver a malicious outcome

authenticate

verify the identity of a user, user device, or other entity, or the integrity of data stored, transmitted, or otherwise exposed to unauthorized modification in an information system, or to establish the validity of a transmission.

authentication

security measure designed to establish the validity of a transmission, message, or originator, or a means of verifying an individual's authorization to receive specific categories of information.

Authentication factor:

Piece of information and/or process used to authenticate or verify the identity of an entity. Authentication factors are divided into four categories: 1) something an entity has (e.g., device signature, passport, hardware device containing a credential, private key); 2) something an entity knows (e.g., password, PIN); 3) something an entity is (e.g., biometric characteristic); or 4) something an entity typically does (e.g., behavior pattern).

authorization

right or a permission that is granted to a system entity to access a system resource. Authorization uses identity attribute, for the subject, and control attributes, for the resource, to decide on a permission. Typically, authorization decisions are based on a policy. An authorization result may be intended for immediate use in accessing a resource, or it may be encoded as the value of an identity attribute and registered with the dentist of the subject for future use.

availability

probability that an asset, under the combined influence of its reliability, maintainability, and security, will be able to fulfill its required function over a stated period of time, or at a given point in time.

border

edge or boundary of a physical or logical security zone.

Bot robot

automated software program used to carry out specific tasks. The word is often used to describe programs, usually run on a server, that automate tasks such as forwarding or sorting e-mail. A bot is also described as a program that operates as an agent for a user or another program or simulates a human activity. On the Internet, the most ubiquitous bots are the programs, also called spiders or crawlers, which access websites and gather their content for search engine indexes.

botnet

collection of software robots, or bots, which run autonomously. A botnet's originator can control the group remotely, possibly for nefarious purposes.

boundary

software, hardware, or other physical barrier that limits access to a system or part of a system.

channel

specific communication link established within a communication conduit.

communication path

logical connection between a source and one or more destinations, which could be devices, physical processes, data items, commands, or programmatic interfaces. The communication path is not limited to wired or wireless networks, but includes other means of communication such as memory, procedure calls, state of physical plant, portable media, and human interactions.

communication security

(1) measures that implement and assure security services in a communication system, particularly those that provide data confidentiality and data integrity and that authenticate communicating entities.

(2) state that is reached by applying security services, in particular, state of data confidentiality, integrity, and successfully authenticated communications entities

compromise

unauthorized disclosure, modification, substitution, or use of information (including plaintext cryptographic keys and other critical security parameters).

conduit

logical grouping of communication assets that protects the security of the channels it contains. This is analogous to the way that a physical conduit protects cables from physical damage.

confidentiality

assurance that information is not disclosed to unauthorized individuals, processes, or devices

control center

central location used to operate a set of assets. Infrastructure industries typically use one or more control centers to supervise or coordinate their operations. If there are multiple control centers (for example, a backup center at a separate site), they are typically connected together via a wide area network. The control center contains the SCADA host computers and associated operator display devices plus ancillary information systems such as a historian.

control equipment

class that includes distributed control systems, programmable logic controllers, SCADA systems, associated operator interface consoles, and field sensing and control devices used to manage and control the process. The term also includes field bus networks where control logic and algorithms are executed on intelligent electronic devices that coordinate actions with each other, as well as systems used to monitor the process and the systems used to maintain the process.

control network

time-critical network that is typically connected to equipment that controls physical processes. The control network can be subdivided into zones, and there can be multiple separate control networks within one company or site.

countermeasure

action, device, procedure, or technique that reduces a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken.

cryptographic algorithm

algorithm based upon the science of cryptography, including encryption algorithms, cryptographic hash algorithms, digital signature algorithms, and key agreement algorithms.

cryptographic key

input parameter that varies the transformation performed by a cryptographic algorithm . Usually shortened to just "key."

cyber attack(s)

Type of attacks where services or applications in the Cyberspace are used or are the target of attack, or where Cyberspace is the source, tool, target, or place of an attack.

data confidentiality

property that information is not made available or disclosed to any unauthorized system entity, including unauthorized individuals, entities, or processes-

data integrity

property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. This term deals with constancy of and confidence in data values, not with the information that the values represent or the trustworthiness of the source of the values.

decryption

process of changing cipher text into plaintext using a cryptographic algorithm and key .

defense in depth

provision of multiple security protections, especially in layers, with the intent to delay if not prevent an attack. Defense in depth implies layers of security and detection, even on single systems, and provides the following features: a) attackers are faced with breaking through or bypassing each layer without being detected; b) flaw in one layer can be mitigated by capabilities in other layers; c) system security becomes a set of layers within the overall network security.

demilitarized zone

perimeter network segment that is logically between internal and external networks. The purpose of a demilitarized zone is to enforce the internal network's policy for external information exchange and to provide external, untrusted sources with restricted access to releasable information while shielding the internal network from outside attacks.

denial of service

prevention or interruption of authorized access to a system resource or the delaying of system operations and functions. In the context of industrial control systems, denial of service can refer to loss of process function, not just loss of data communications.

digital signature

result of a cryptographic transformation of data which, when properly implemented, provides the services of origin authentication, data integrity, and signer non-repudiation.

distributed control system

type of control system in which the system elements are dispersed but operated in a coupled manner. Distributed control systems may have shorter coupling time constants than those typically found in SCADA systems.

domain

environment or context that is defined by a security policy, security model, or security architecture to include a set of system resources and the set of system entities that have the right to access the resources.

eavesdropping

monitoring or recording of communicated information by unauthorized parties.

electronic security

actions required to preclude unauthorized use of, denial of service to, modifications to, disclosure of, loss of revenue from, or destruction of critical systems or informational assets. Electronic security includes the concepts of identification, authentication, accountability, authorization, availability, and privacy.

encryption

cryptographic transformation of plaintext into cipher text that conceals the data’s original meaning to prevent it from being known or used. If the transformation is reversible, the corresponding reversal process is called "decryption," which is a transformation that restores encrypted data to its original state.

enterprise

business entity that produces or transports products or operates and maintains infrastructure services.

enterprise system

collection of information technology elements (i.e., hardware, software and services) installed with the intent to facilitate an organization’s business process or processes (administrative or project).

equipment under control

equipment, machinery, apparatus or plant used for manufacturing, process, transportation, medical or other activities.

field I/O network

communications link (wired or wireless) that connects sensors and actuators to the control equipment.

firewall

inter network connection device that restricts data communication traffic between two connected networks. A firewall may be either an application installed on a general purpose computer or a dedicated platform (appliance) that forwards or rejects/drops packets on a network. Typically firewalls are used to define zone borders. Firewalls generally have rules restricting which ports are open.

gateway

relay mechanism that attaches to two (or more) computer networks that have similar functions but dissimilar implementations and that enables host computers on one network to communicate with hosts on the other. Also described as an intermediate system that is the translation interface between two computer networks.

geographic site

subset of an enterprise's physical, geographic, or logical group of assets. A geographic site may contain areas, manufacturing lines, process cells, process units, control centers, and vehicles and may be connected to other sites by a wide area network.

guard

gateway that is interposed between two networks (or computers or other information systems) operating at different security levels (one network is usually more secure than the other) and is trusted to mediate all information transfers between the two networks, either to ensure that no sensitive information from the more secure network is disclosed to the less secure network, or to protect the integrity of data on the more secure network .

host

computer that is attached to a communication sub network or inter network and can use services provided by the network to exchange data with other attached systems.

industrial control systems

humans, hardware and software that can affect or influence the safe, secure, and reliable operation of an industrial process. These systems include, but are not limited to: a) industrial control systems, including distributed control systems (DCSs), programmable logic controllers (PLCs), remote terminal units (RTUs), intelligent electronic devices, supervisory control and data acquisition (SCADA), networked electronic sensing and control, and monitoring and diagnostic systems. (In this context, process control systems include basic process control system and safety instrumented system (SIS) functions, whether they are physically separate or integrated.) b) associated information systems such as advanced or multivariable control, online optimizers, dedicated equipment monitors, graphical interfaces, process historians, manufacturing execution systems, and plant information management systems. c) associated internal, human, network, or machine interfaces used to provide control, safety, and manufacturing operations functionality to continuous, batch, discrete, and other processes.

initial risk

risk before controls or countermeasures have been applied.

Insider

trusted person, employee, contractor, or supplier who has information that is not generally known to the public.

integrity

quality of a system reflecting the logical correctness and reliability of the operating system, the logical completeness of the hardware and software implementing the protection mechanisms, and the consistency of the data structures and occurrence of the stored data.

interception

capture and disclosure of message contents or use of traffic analysis to compromise the confidentiality of a communication system based on message destination or origin, frequency or length of transmission, and other communication attributes.

interface

logical entry or exit point that provides access to the module for logical information flows.

intrusion

unauthorized act of compromising a system.

intrusion detection

security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner.

IP address

address of a computer or device that is assigned for identification and communication using the Internet Protocol and other protocols.

key management

process of handling and controlling cryptographic keys and related material (such as initialization values) during their life cycle in a cryptographic system, including ordering, generating, distributing, storing, loading, escrowing, archiving, auditing, and destroying the keys and related material.

local area network

communications network designed to connect computers and other intelligent devices in a limited geographic area (typically less than 10 kilometers).

malicious code

programs or code written for the purpose of gathering information about systems or users, destroying system data, providing a foothold for further intrusion into a system, falsifying system data and reports, or providing time-consuming irritation to system operations and maintenance personnel. Malicious code attacks can take the form of viruses, worms, Trojan Horses, or other automated exploits. Malicious code is also often referred to as malware.

non repudiation

security service that provides protection against false denial of involvement in a communication.

OPC

set of specifications for the exchange of information in a process control environment. The abbreviation OPC originally came from OLE for Process Control, where OLE was short for Object Linking and Embedding.

outsider

person or group not trusted with inside access, who may or may not be known to the targeted organization.

penetration

successful unauthorized access to a protected system resource.

phishing

type of security attack that lures victims to reveal information, by presenting a forged email to lure the recipient to a web site that looks like it is associated with a legitimate source.

plaintext

unencoded data that is input to and transformed by an encryption process, or that is output by a decryption process.

privilege

authorization or set of authorizations to perform specific functions, especially in the context of a computer operating system. Examples of functions that are controlled through the use of privilege include acknowledging alarms, changing set points, modifying control algorithms.

process

series of operations performed in the making, treatment or transportation of a product or material.

protocol

set of rules (i.e., formats and procedures) to implement and control some type of association (e.g., communication) between systems.

reference model

structure that allows the modules and interfaces of a system to be described in a consistent manner.

reliability

ability of a system to perform a required function under stated conditions for a specified period of time.

remote access

use of systems that are inside the perimeter of the security zone being addressed from a different geographical location with the same rights as when physically present at the location.

remote client

asset outside the control network that is temporarily or permanently connected to a host inside the control network via a communication link in order to directly or indirectly access parts of the control equipment on the control network.

repudiation

denial by one of the entities involved in a communication of having participated in all or part of the communication.

residual risk

the remaining risk after the security controls or countermeasures have been applied.

risk

expectation of loss expressed as the probability that a particular threat will exploit a particular vulnerability with a particular consequence.

risk assessment

process that systematically identifies potential vulnerabilities to valuable system resources and threats to those resources, quantifies loss exposures and consequences based on probability of occurrence, and (optionally) recommends how to allocate resources to countermeasures to minimize total exposure. Types of resources include physical, logical and human. Risk assessments are often combined with vulnerability assessments to identify vulnerabilities and quantify the associated risk. They are carried out initially and periodically to reflect changes in the organization's risk tolerance, vulnerabilities, procedures, personnel and technological changes.

risk management

process of identifying and applying countermeasures commensurate with the value of the assets protected based on a risk assessment .

risk mitigation controls

combination of countermeasures and business continuity plans.

role-based access control

form of identity-based access control where the system entities that are identified and controlled are functional positions in an organization or process.

router

gateway between two networks at OSI layer 3 and that relays and directs data packets through that inter-network. The most common form of router passes Internet Protocol (IP) packets.

safety

freedom from unacceptable risk .

safety network

network that connects safety-instrumented systems for the communication of safety related information.

secret

condition of information being protected from being known by any system entities except those intended to know it .

security

- measures taken to protect a system.
- condition of a system that results from the establishment and maintenance of measures to protect the system.
- condition of system resources being free from unauthorized access and from unauthorized or accidental change, destruction, or loss.
- capability of a computer-based system to provide adequate confidence that unauthorized persons and systems can neither modify the software and its data nor gain access to the

system functions, and yet to ensure that this is not denied to authorized persons and systems.

– prevention of illegal or unwanted penetration of or interference with the proper and intended operation of an industrial automation and control system.

security architecture

plan and set of principles that describe the security services that a system is required to provide to meet the needs of its users, the system elements required to implement the services, and the performance levels required in the elements to deal with the threat environment. Security architecture would be an architecture to protect the control network from intentional or unintentional security events.

security audit

independent review and examination of a system's records and activities to determine the adequacy of system controls, ensure compliance with established security policy and procedures, detect breaches in security services, and recommend any changes that are indicated for countermeasures.

security components

assets such as firewalls, authentication modules, or encryption software used to improve the security performance of an industrial automation and control system.

security event

occurrence in a system that is relevant to the security of the system.

security function

function of a zone or conduit to prevent unauthorized electronic intervention that can impact or influence the normal functioning of devices and systems within the zone or conduit.

security incident

adverse event in a system or network or the threat of the occurrence of such an event.

security intrusion

security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system (or system resource) without having authorization to do so.

security level

level corresponding to the required effectiveness of countermeasures and inherent security properties of devices and systems for a zone or conduit based on assessment of risk for the zone or conduit.

security objective

aspect of security which to achieve is the purpose and objective of using certain mitigation measures, such as confidentiality, integrity, availability, user authenticity, access authorization, accountability.

security perimeter

boundary (logical or physical) of the domain in which a security policy or security architecture applies, i.e., the boundary of the space in which security services protect system resources.

security performance

program's compliance, completeness of measures to provide specific threat protection, post compromise analysis, review of changing business requirements, new threat and vulnerability information, and periodic audit of control systems to ensure security measures remain effective and appropriate. Tests, audits, tools, measures, or other methods are required to evaluate security practice performance.

security policy

set of rules that specify or regulate how a system or organization provides security services to protect its assets.

security procedures

definitions of exactly how practices are implemented and executed. Security procedures are implemented through personnel training and actions using currently available and installed technology.

security program

a combination of all aspects of managing security, ranging from the definition and communication of policies through implementation of best industry practices and ongoing operation and auditing.

security services

mechanisms used to provide confidentiality, data integrity, authentication, or no repudiation of information.

security violation

act or event that disobeys or otherwise breaches security policy through an intrusion or the actions of a well-meaning insider.

security zone

grouping of logical or physical assets that share common security requirements. A zone has a clear border with other zones. The security policy of a zone is typically enforced by a combination of mechanisms both at the zone edge and within the zone. Zones can be hierarchical in the sense that they can be comprised of a collection of subzones.

sensors and actuators

measuring or actuating elements connected to process equipment and to the control system.

server

device or application that provides information or services to client applications and devices.

spoof

pretending to be an authorized user and performing an unauthorized action.

supervisory control and data acquisition (SCADA) system

type of loosely coupled distributed monitoring and control system commonly associated with electric power transmission and distribution systems, oil and gas pipelines, and water and sewage systems.

system software

special software designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system and associated programs and data.

threat

potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm.

threat action

assault on system security.

traffic analysis

inference of information from observable characteristics of data flow(s), even when the data are encrypted or otherwise not directly available, including the identities and locations of source(s) and destination(s) and the presence, amount, frequency, and duration of occurrence.

use case

technique for capturing potential functional requirements that employs the use of one or more scenarios that convey how the system should interact with the end user or another system to achieve a specific goal. Typically use cases treat the system as a black box, and the interactions with the system, including system responses, are as perceived from outside of the system. Use cases are popular because they simplify the description of requirements, and avoid the problem of making assumptions about how this functionality will be accomplished.

user

person, organization entity, or automated process that accesses a system, whether authorized to do so or not.

vulnerability

flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's integrity or security policy.

wide area network

communications network designed to connect computers, networks and other devices over a large distance, such as across the country or world.

wiretapping

attack that intercepts and accesses data and other information contained in a flow in a communication system. Although the term originally referred to making a mechanical connection to an electrical conductor that links two nodes, it is now used to refer to reading information from any sort of medium used for a link or even directly from a node, such as a gateway or sub network switch. "Active wiretapping" attempts to alter the data or otherwise affect the flow; "passive wiretapping" only attempts to observe the flow and gain knowledge of information it contains.

1.5 Acronym and symbols

Terminology	Description
ACL	Access Control List
ARP	Address Resolution Protocol
CIA	Confidentiality, Integrity, and Availability
CERT	Computer Emergency Response Team
COTS	Commercial off the Shelf
DCS	Distributed Control System
DDoS	Distributed Denial of Service
DMZ	Demilitarized Zone
DoS	Denial of Service
FTA	Fault Tree Analysis
ICT	Information and Communication Technologies
IDS	Intrusion Detection System
IP	Internet Protocol
IRP	Integrated Risk Prediction

LAN	Local Area Network
MIS	Management Information System
NIST	National Institute of Standards and Technology
OLE	Object Linking and Embedding
OPC	OLE for Process Control
OSI	Open Systems Interconnect
PLC	Programmable Logic Controller
QoS	Quality of Service
RAT	Remote Access Trojan
RTOS	Real-Time Operating System
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
STP	Spanning Tree Protocol
UDP	User Datagram Protocol
WAN	Wide Area Network

2 Malware classification

National Institute of Standards and Technology (NIST) in [2] gives a complete and compact view of malware as follows.

Malware, also known as malicious code and malicious software, refers to a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system or otherwise annoying or disrupting the victim.

Malware has become the most significant external threat to most systems, causing widespread damage and disruption, and necessitating extensive recovery efforts within most organizations. Organizations also face threats that are often associated with malware. One of these forms that has become commonplace is phishing, which is using deceptive computer-based means to trick individuals into disclosing sensitive information. Another common form is virus hoaxes, which are false warnings of new malware threats.

This section addresses several major forms of malware, including viruses, worms, Trojan horses, malicious mobile code, blended attacks, spyware tracking cookies, and attacker tools such as backdoors and rootkits.

Both malware and the defenses against malware continue to evolve, each in response to improvements in the other. As a new category of threats becomes more serious, organizations should plan and implement appropriate controls to mitigate it. Awareness of new and emerging threats and protective capabilities should be part of efforts to prevent malware incidents.

2.1 Viruses

A virus is characterized by the fact that self-replicates by inserting copies of itself into host programs or data files. Viruses are often triggered through user interaction, such as opening a file or running a program. Viruses can be divided into the following two subcategories:

- **Compiled Viruses**. A compiled virus is executed by an operating system. Types of compiled viruses include file infector viruses, which attach themselves to executable programs; boot sector viruses, which infect the master boot records of hard drives or the boot sectors of removable media; and multipartite viruses, which combine the characteristics of file infector and boot sector viruses.

- **Interpreted Viruses**. Interpreted viruses are executed by an application. Within this subcategory, macro viruses take advantage of the capabilities of applications. Macro programming language to infect application documents and document templates, while scripting viruses infect scripts that are understood by scripting languages processed by services on the OS.

2.2 Worms

A worm is a self-replicating, self-contained program that usually executes itself without user intervention. Worms are divided into two categories:

- **Network Service Worms.** A network service worm takes advantage of vulnerability in a network service to propagate itself and infect other systems.

- **Mass Mailing Worms**. A mass mailing worm is similar to an e-mail -borne virus but is self-contained, rather than infecting an existing file.

2.3 Trojan Horses

A Trojan horse is a self-contained, non replicating program that, while appearing to be benign, actually has a hidden malicious purpose. Trojan horses either replace existing files with malicious versions or add new malicious files to systems. They often deliver other attacker tools to systems.

2.4 Malicious Mobile Code

Malicious mobile code is software with malicious intent that is transmitted from a remote system to a local system and then executed on the local system, typically without the user's explicit instruction. Popular languages for malicious mobile code include Java, ActiveX, JavaScript, and VBScript.

2.5 Tracking Cookies

A tracking cookie is a persistent cookie that is accessed by many Web sites, allowing a third party to create a profile of a user's behaviour. Tracking cookies are often used in conjunction with Web bugs, which are tiny graphics on Web sites that are referenced within the HTML content of a Web page or e-mail. The only purpose of the graphic is to collect information about the user viewing the content.

2.6 Blended Attacks

A blended attack uses multiple infection or transmission methods. For example, a blended attack could combine the propagation methods of viruses and worms.

2.7 Attacker Tools

Various types of attacker tools might be delivered to a system as part of a malware infection or other system compromise. These tools allow attackers to have unauthorized access to or use of infected systems and their data, or to launch additional attacks. Popular types of attacker tools are as follows:

- **Backdoors**. A backdoor is a malicious program that listens for commands on a certain TCP or UDP port. Most backdoors allow an attacker to perform a certain set of actions on a system, such as acquiring passwords or executing arbitrary commands. Types of backdoors include zombies (also known as bots), which are installed on a system to cause it to attack other systems, and remote administration tools, which are installed on a system to enable a remote attacker to gain access to the system's functions and data as needed.

- **Keystroke Loggers**. A keystroke logger monitors and records keyboard use. Some require the attacker to retrieve the data from the system, whereas other loggers actively transfer the data to another system through e-mail, file transfer, or other means.

- **Rootkits.** A rootkit is a collection of files that is installed on a system to alter its standard functionality in a malicious and stealthy way. A rootkit typically makes many changes to a system to hide the rootkit's existence, making it very difficult to determine that the rootkit is present and to identify what the rootkit has changed.

- **Web Browser Plug-Ins**. A Web browser plug-in provides a way for certain types of content to be displayed or executed through a Web browser. Attackers often create malicious Web browser plug-ins that act as spyware and monitor all use of the browser.

- **E-Mail Generators**. An e-mail generating program can be used to create and send large quantities of e-mail, such as malware, spyware, and spam, to other systems without the user's permission or knowledge.

- **Attacker Toolkits**. Many attackers use toolkits containing several different types of utilities and scripts that can be used to probe and attack systems, such as packet sniffers, port scanners, vulnerability scanners, password crackers, remote login programs, and attack programs and scripts.

In addition to malware, there are also a few common non-malware threats that are often associated with malware. Phishing uses computer-based means to trick users into revealing sensitive (i.e. financial) information and data. Phishing attacks frequently place malware or attacker tools on systems. An additional malicious content threat is virus hoaxes - false warnings of new malware threats.

3 Stealing information and starting an attack

The techniques used for steal information and start a cyber attack to a SCADA system is not too different to an ICT system. Following, there is an example of typical attack phases:

1. Password guessing
2. Port scanning
3. Exploitation
4. Man-In-The-Middle (MITM)
5. Denial of Service (DoS)

3.1 Password guessing

An attacker that tries to find a password can work in two ways:

- dictionary
- brute-force

the first one simply tries all the words in a set. This set can be big enough to include special characters and lower and upper case. This method can be very efficient due to the fact that usually the passwords are simply actual words. The second one, the brute-force attack, tries all the combination of characters of a certain length. If the length is unknown, usually it tries from a min to a max number of characters. Due to the high complexity of the combinatorial problem, this method is very inefficient when the length and the sets of characters grow.

3.2 Port Scanning

A port scanning consists in sending client requests to a range of server port addresses on a host, with the goal of finding an active port and exploiting a known vulnerability of that service [(http://tools.ietf.org/html/ rfc2828)]

3.3 Exploitation

The term exploitation refers to the act of successfully making an attack (i.e. Denial of Service (DoD)) on a computer system, taking advantage of a particular vulnerability that the system offers to the intruders.

3.4 Man In The Middle

The MITM attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. A MITM attack can succeed only when the attacker can impersonate each endpoint to the satisfaction of the other; it's an attack on mutual authentication. All the communications are subjected to eavesdropping, it all depends on how smart is the attacker. Technically, even an encryption without the exchange of the encryption key over the communication link can be considered unsecure; in fact, a social engineering attacker is capable to obtain locally the key. The best solution is to use an algorithm that takes time to decrypt/crypt the message, but it can't be used if the application is time critical. Social engineering techniques [3] are based on specific attributes of human decision making known as cognitive biases. With social engineering (such as pretexting, phishing, Interactive Voice Response (IVR) or phone phishing, baiting), one can deceive a person by tricking him/her into supplying personal information and passwords. Any method of communication can be used to perpetrate this fraud. Using viruses or downloading files which have Backdoor or Trojan horses within, if the user of a remote management tool has been infected or tries to place the backdoor or Trojan horse which executes tasks similar to Back Orifice, Net bus, Netcat and Key Logger, in most cases, those become shut-off by a virus vaccine or security tool. However, new types of these cannot be blocked. That is, there are not many means of defence if perpetrators try to make an attack with new viruses and hacking patterns after setting a target. It would not be difficult to acquire information about access codes and passwords using Backdoor or a Trojan horse. If things have already progressed up to this point, the control power of the control system will be handed over to the organizations or users with malicious.

3.5 Denial of Service

A DoS attack or Distributed DoS (DDoS) attack is an attempt to make a computer resource unavailable to its intended uses. Although the means to carry out, motives for, and targets of DoS attack may vary, it generally consists of the concerted efforts of a person, or multiple people to prevent an internet site or service from functioning efficiently or at all, temporarily or indefinitely. One common method of attack involves saturating the target machine with external communication requests, such that it cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable.

In Figure 3.1 there's an example of DDoS attack.

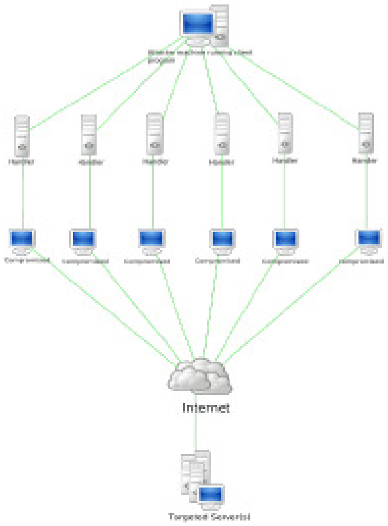


Figure 3-1 DDoS attack

The attacker, to perform the DoS attack, first infects a set of machines, called zombies or bots (needed to meet the problems of different values of upload and download bandwidth). When the attacker obtains the control of the machines, checks if they are online, and if a certain number of zombies are online, then starts the DDoS attack, launching several requests (like SYN or PING) that require from the victim to send an ACK and make the buffer of the victim's router in overflow. Once in overflow, the devices can fail in two ways: fail open and fail close (like a open circuit or an short circuit). In one case, a malicious package could pass, in the other case, the net could be considered like broken in one point, and the problems associated to this event are only about topology and then connectivity.

4 Industrial Control System 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 4.1 shows a hierarchy of logical levels, proposed by 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.

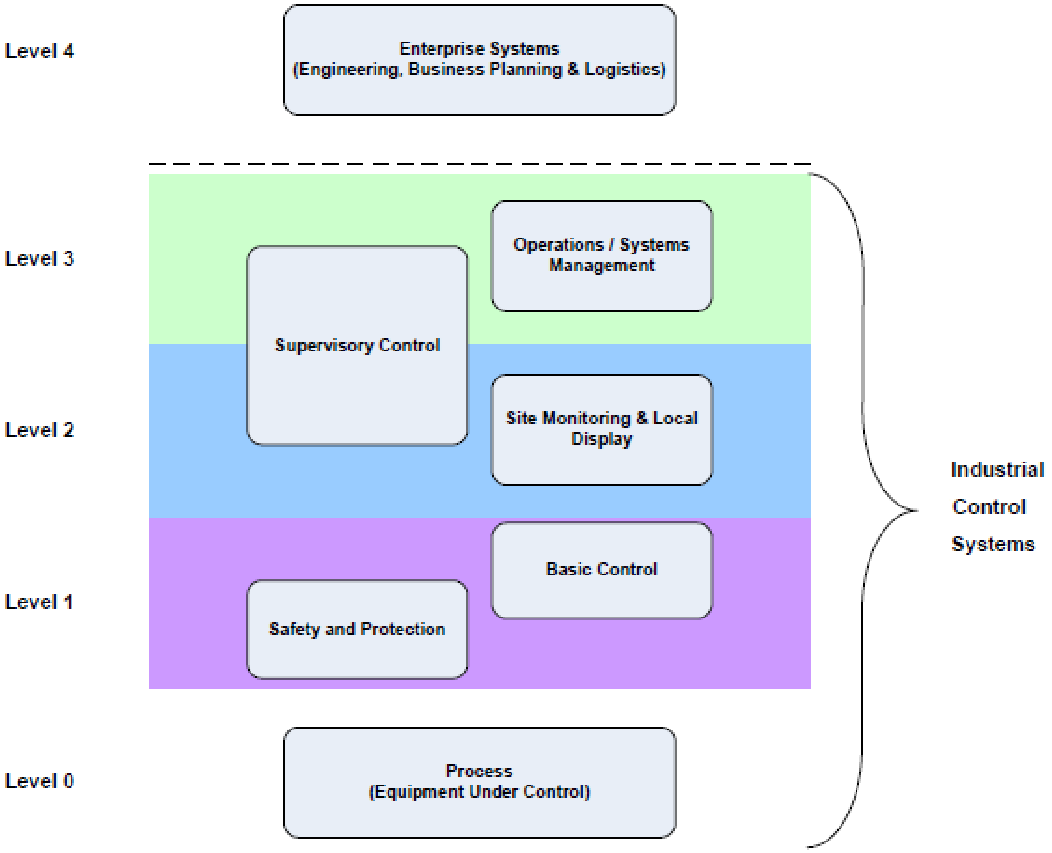

Level 4
Enterprise Systems
(Engineering, Business Planning & Logistics)
Level 3
Operations / Systems Management
Supervisory Control
Level 2
Site Monitoring & Local Display
Level 1
Basic Control
Safety and Protection
Level 0
Process
(Equipment Under Control)
Industrial Control Systems

Figure 4-1 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
- RTUs.

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 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.

4.1 Comparing ICS and ICT systems

With reference to figure 4.1 the level 4, enterprise system relies on ICT technologies. In case of Critical Infrastructures, the enterprise system is constituted by the enterprise/corporate network.

A synthesis of the major differences that arise by comparing Industrial Control System (ICS)/SCADA and Information and Communication Technology (ICT) of the enterprise/corporate network, when considering security for ICS, is reported in the following [1]:

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, (eg. 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, 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 very 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 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.

5 ICS cyber threats, vulnerabilities and attacks

Initially [1], 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 these systems.

Figure 5.1 [4] shows an isolated ICS and figure 5.2 [4] shows an integrated ICS with a corporate network.

Particularly, in figure 5.1 differently from figure 5.2, 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.

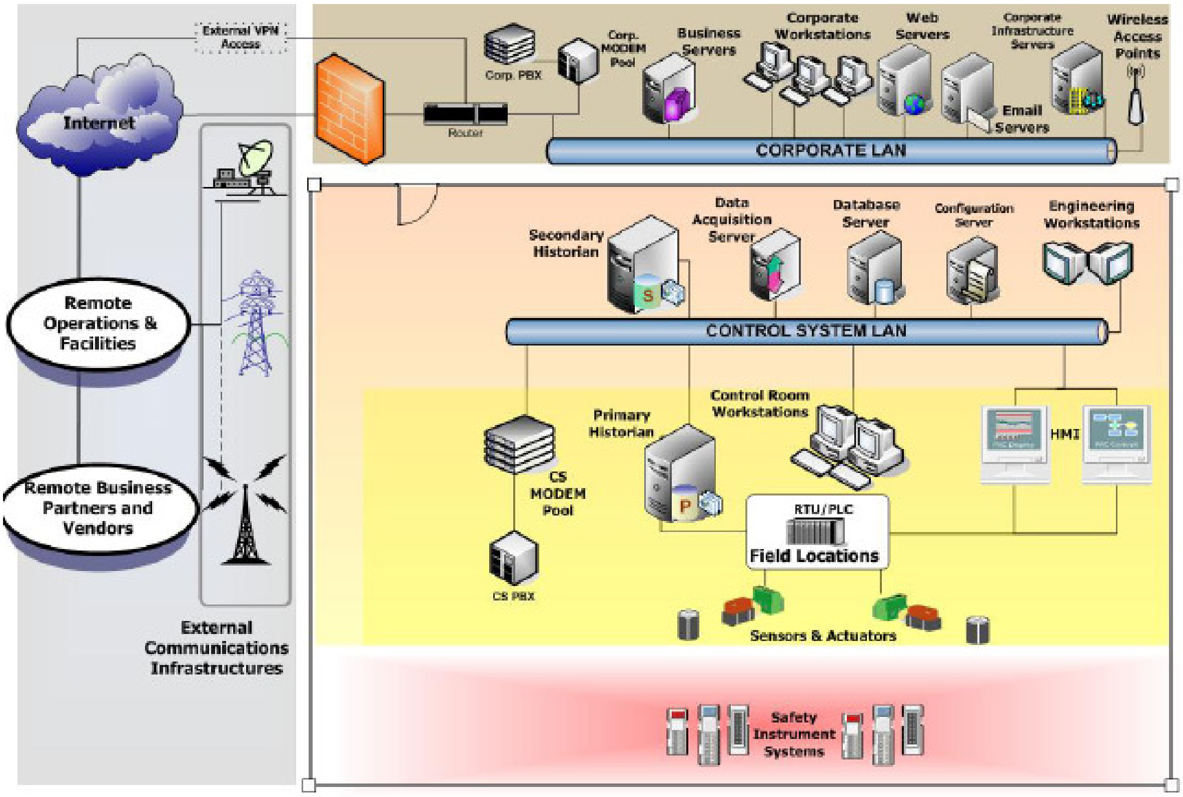


Figure 5-1 Isolated ICS [4]

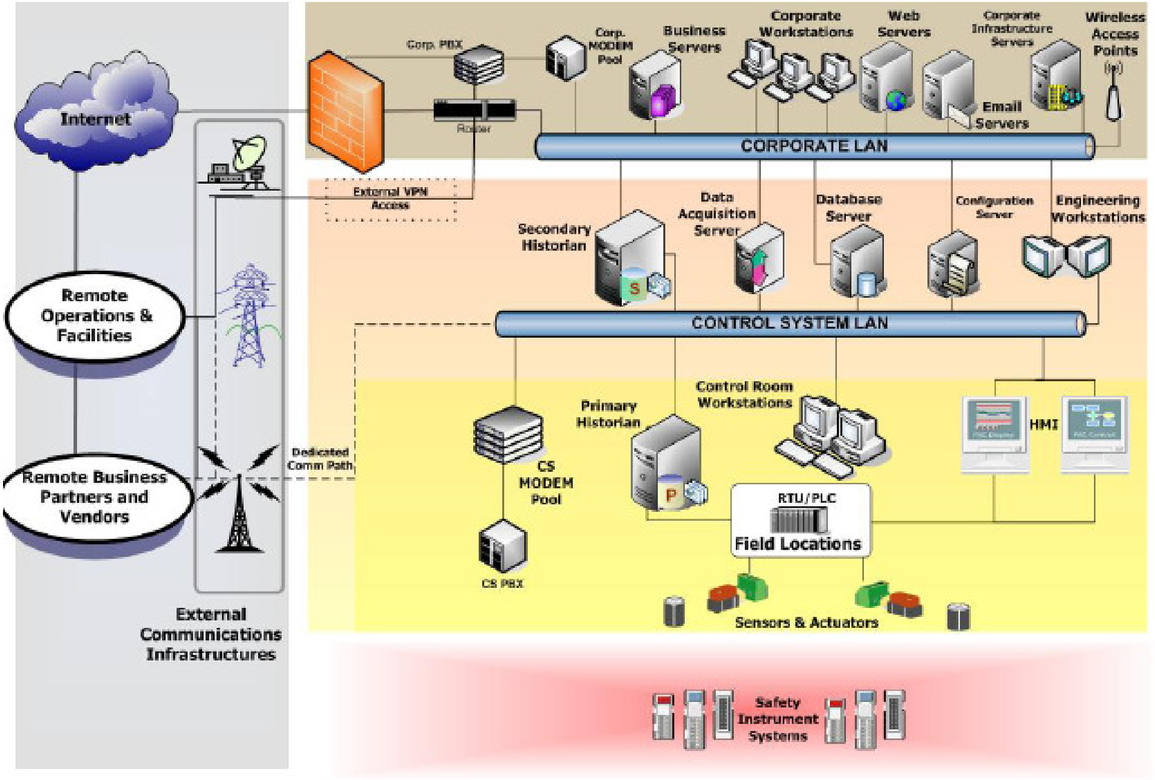


Figure 5-2 Integrated ICS and corporate network [4]

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 are commonly provided in the non-ICS world by in-device SSL or other cryptographic techniques [5].

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

[6]. Additionally, application whitelisting solutions[1] 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 AES encryption rather than replacing all existing nodes.

By this method an hardware device that provides 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 5.3. These devices will then intercept outgoing datagrams and add IPSec protection to them, and strip it off incoming datagrams.

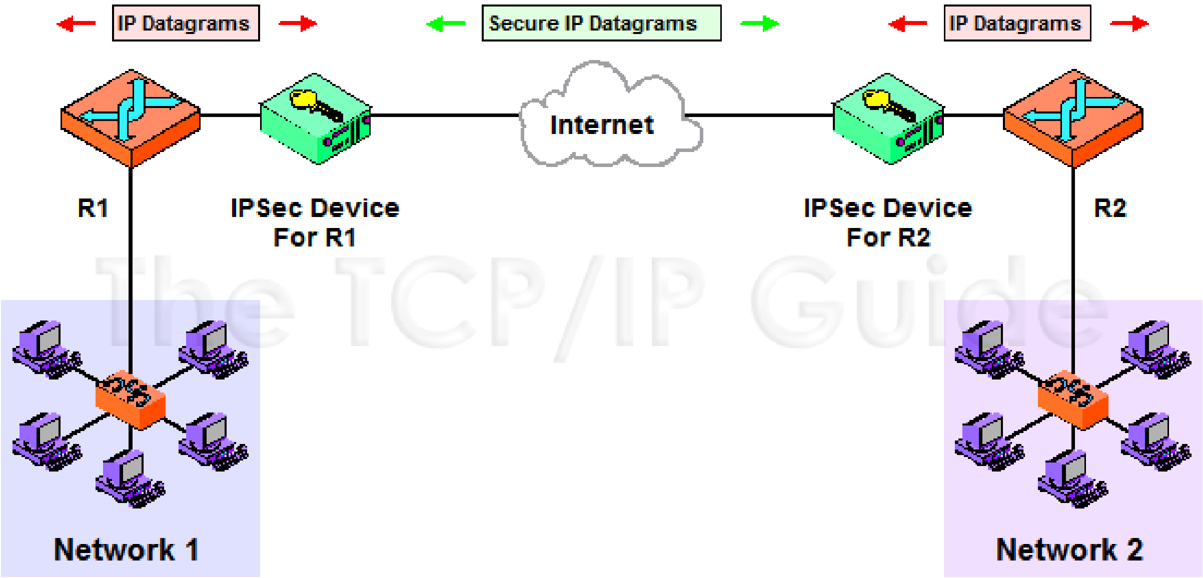


Figure 5-3 IPSec "Bump In The Wire" (BITW) Architecture [6]

Critical Infrastructure cyber vulnerabilities involve the enterprise/corporate network, the industrial control systems and the critical infrastructure itself (i.e. power system). Figure 5.4, where Industrial Control systems are part of Process Control Network, and figure 5.5, where ICS is specified as SCADA, show how they are linked together. Both figures 5.4 and 5.5 come from [7]. Figure 5.5 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

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

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.

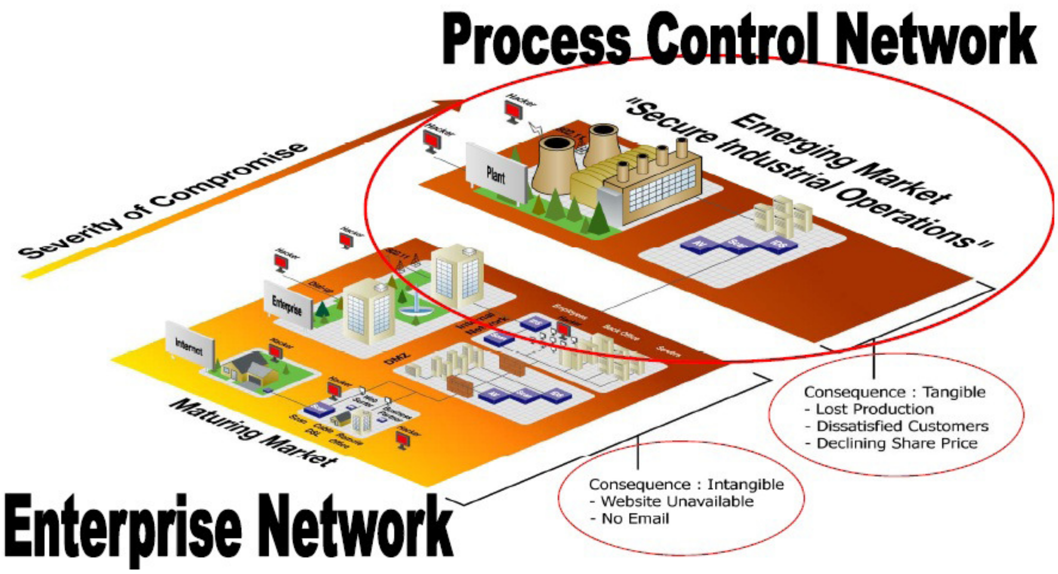


Figure 5-4 External view with the enterprise network linked with the process control network [4]

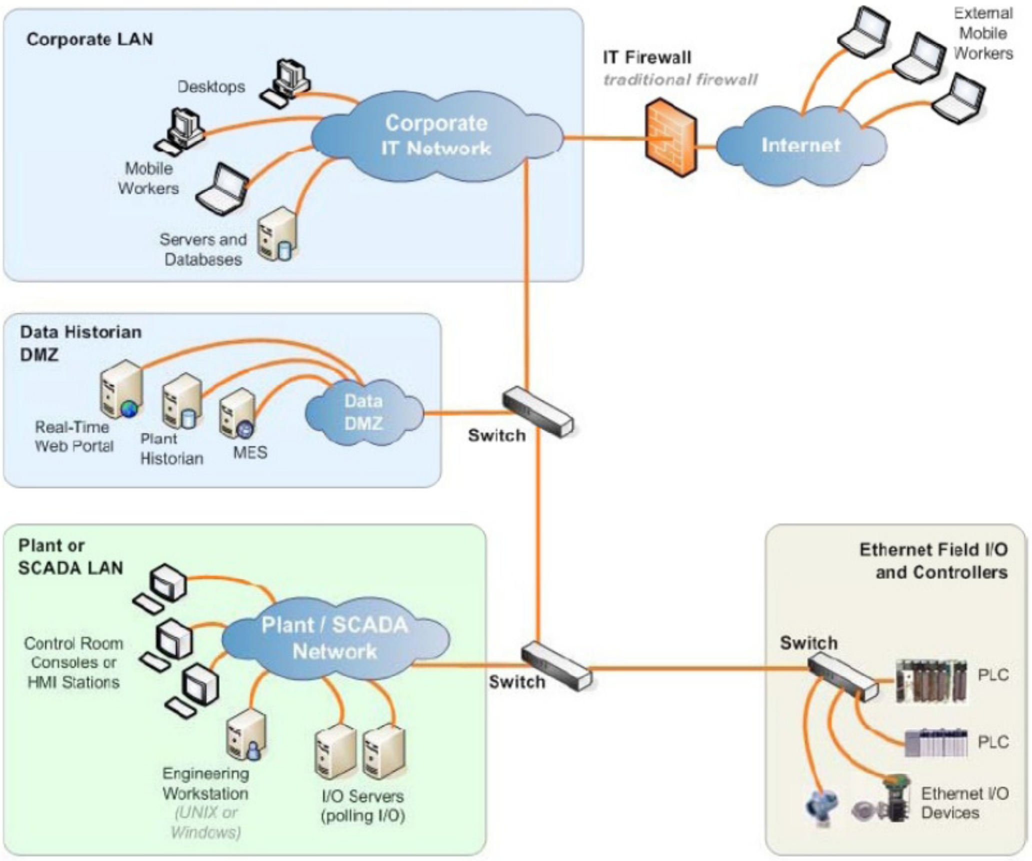


Figure 5-5 Component view of corporate network and SCADA [4]

5.1 Vulnerabilities

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 [8,9,10].

- 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.

5.2 Threats

Due to the above vulnerabilities, there may be security threats [8,11]. 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 MIS servers or PCs. Even though 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 constructed and delivered, sometimes, the remote access for its maintenance by the vendor 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.

5.3 Attack sources and attacker profiles

5.3.1 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 attackers 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 [13]:

1. Internal
   a. Non malicious: employees or contractors causing unintentional damage
   b. Malicious: system users with extensive internal knowledge of the system who intentionally cause damage
2. External
   a. Opportunistic: hackers seeking a challenge
   b. Deliberate: malicious, well-funded political activists, organized crime groups, or nation states

According to the above classification, following there are related examples of historical attacks [13]

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 system's central monitoring computers. His tampering resulted in 800,000 liters 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 Advanced Persistent Threats (APT), section 5.5.

5.3.2 Attacker profiles

There are many attacker profiles:

The lone individual (coder/hacker), or small group of individuals. The threat behind any cyber-attack is a human who has access to a computer and the internet.

A highly skilled coder is a sophisticated programmer who has the ability to find unique vulnerabilities in existing software and to create working exploit codes. They would have the equivalent of an undergraduate degree in computer science with an emphasis on the systems area. They would have a deep understanding of the TCP/IP network protocol as well as network and security protocols in general, and understand operating systems concept. They would need several years of hands-on experience in an IT environment so they could perform host platform vulnerability assessments and understand hardening standards and methodologies [14].

The low skill coder, often called the "script kiddie", is the most common type of hacker. Their name (script kiddie) comes from the fact that members of this group generally rely on previously coded scripts and pre-packaged hacking tools downloaded from the Internet to do their hacking. Script kiddies are often challenged by the notion of gaining unauthorized access and are sometimes open to using untested pieces of code without knowing their consequences. If a low skill coder penetrates a corporate network, and have malicious intent, they could wreak havoc until they are detected. A low skill coder would be subject to quick detection because of their inability to cover their tracks [14].

There are mid skill coders who have capabilities in between the low skill coder and the highly skilled coder but we usually focus on the highly skilled coder because of their capabilities to actually impact systems, and the low skill coders because they make up the overwhelming majority of the "hackers" in the world [14].

It is important to note that most highly skilled coders/hackers are not malicious. In fact, some are actively involved in developing technologies that can be used to improve overall computer and network security. Coders can work independently or through a network of hacking teams that run exploits from a variety of locations, making it difficult to trace the activities back to their source. These teams can be developed in Internet Relay Chat (IRC) channels, in conferences such as DefCon, or in small groups of computer savvy friends. Often coders create the programs and other members of the team run them against target networks. This creates a reputation for the group rather than a single individual [14].

The Insider(s). The disgruntled insider is a principal source of computer crime and sabotage. Insiders may not need a great deal of knowledge about computer intrusions because their knowledge of a target system often allows them to gain unrestricted access to cause damage to the system or to steal system data. The insider threat also includes outsourcing vendors as well as employees who accidentally introduce malware into systems. Insiders may be employees, contractors, or business partners [14].

Criminal Groups. The primary motivation of a criminal group launching, or seeking to launch, a cyber-attack on a SCADA facility would be extortion. In [14] is revealed that a CIA official publicly announced that hackers have penetrated power systems in several regions outside the United States, and in at least one case caused a power outage affecting multiple cities.

Terrorist Group. Most terrorist groups seek higher-impact targets than bringing down a critical infrastructure, even one in the USA. However, a group with a long enough time horizon and enough financial backing may develop capabilities on par with nation-states.

Nation-States. A nation state, or highly motivated terrorist group, most likely could develop the capabilities to bring down a SCADA facility, or even a network of facilities. Besides being able to recruit highly-skilled coders, hire control system engineers and bribe insiders, they also have the capabilities to do the following:

- obtain the source code for proprietary software and thus identify vulnerabilities unknown to the general public
- persuade vendors or their employees to intentionally insert "backdoors" or other zero-day vulnerabilities into their software code or hardware devices. A zero-day vulnerability is a vulnerability which the adversary has known about for some time but the defender has known about for zero days
- obtain (usually buy) the system of interest in order to understand its operational strengths and weaknesses as well as its vulnerabilities

Raoul Chiesa in [15] has profiled the “new” kind of hacker. His work can be summarized in four tables with a detailed analysis and correlation of profiles. One of them, table 5.1, is following reported as an example

Table 5-1 Hacker profiles

PROFILE	RANK	IMPACT LEVEL	TARGET
Wanna be lamer	Amateur	Null	End-user
Script kiddie	Amateur	Low	SME| Specific security flows
Cracker	Hobbiest	Medium| High Business	company
Ethical hacker	Hobbiest	Medium	Vendor| Technology
Quit,Pranoid skilled hacker	Hobbiest	Medium| High	On necessary
Cyber Warrior	Professional	High	“Symbol” business company| End-user
Industrial spy	Professional	High	Business company | Corporation
Governament agent	Professional	High	Governament | Suspected Terrorist |Strategic Company | Individual
Military hacker	Professional	High	Governament | Strategic company

5.4 Typical attacks

5.4.1 Targeted and flood based cyber 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, defense 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 to be studied in greater detail.

5.4.2 Attacks on the Communication Stack

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 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 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 an host connected to different actuators, then physical disasters of different scales are initiated. Static 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 [16] 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: is a de facto standard of application layer protocol used in industrial networks. The lack of encryption or any other security measures of MODBUS exposes this protocol to different vulnerabilities which have been analyzed in [17]. 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: is a set of communications protocols used between components in process automation systems specifically designed for use in SCADA applications [18]. 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: 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 protocol presents different vulnerabilities (). An example is the opportunistic DoS attack [19] 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 attacks 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 [20] implementation of the 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.

5.5 Advanced Persistent Threats

Advanced persistent threat (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.

Within the computer security community, and increasingly within the media, the term is almost always used in reference to a long-term pattern of sophisticated hacking attacks aimed at governments, companies, and political activists, and by extension, also to refer to the groups behind these attacks. A common misconception associated with the APT is that the APT only targets Western governments. While examples of technological APTs against Western governments may be more publicized in the West, actors in many nations have used the technological (cyber) APT as a means to gather intelligence on individuals and groups of individuals of interest. The United States Cyber Command is tasked with coordinating the US military's response to this cyber threat.

Numerous sources have alleged that some APT groups are affiliated with, or are agents of, nation-states.

Definitions of precisely what an APT is can vary, but can be summarized by their named requirements below:

Advanced – Operators behind the threat have a full spectrum of intelligence-gathering techniques at their disposal. These may include computer intrusion technologies and techniques, but also extend to conventional intelligence-gathering techniques such as telephone-interception technologies and satellite imaging. While individual components of the attack may not be classed as particularly “advanced” (e.g. malware components generated from commonly available do-it-yourself malware construction kits, or the use of easily procured exploit materials), their operators can typically access and develop more advanced tools as required. They often combine multiple targeting methods, tools, and techniques in order to reach and compromise their target and maintain access to it. Operators may also demonstrate a deliberate focus on operational security that differentiates them from "less advanced" threats.

Persistent – Operators give priority to a specific task, rather than opportunistically seeking information for financial or other gain. This distinction implies that the attackers are guided by external entities. The targeting is conducted through continuous monitoring and interaction in order to achieve the defined objectives. It does not mean a barrage of constant attacks and malware updates. In fact, a “low-and-slow” approach is usually more successful. If the operator loses access to their target they usually will reattempt access, and most often, successfully. One of the operator's goals is to maintain long-term access to the target, in contrast to threats who only need access to execute a specific task.

Threat – APTs are a threat because they have both capability and intent. APT attacks are executed by coordinated human actions, rather than by mindless and automated pieces of code. The operators have a specific objective and are skilled, motivated, organized and well funded.

Following there is a short description of the most powerful, systematic and popular APT.

5.5.1 Stuxnet

Unlike most malware, Stuxnet [21] 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 [21]:

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, that 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 5.6 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 choose a new process to infect.

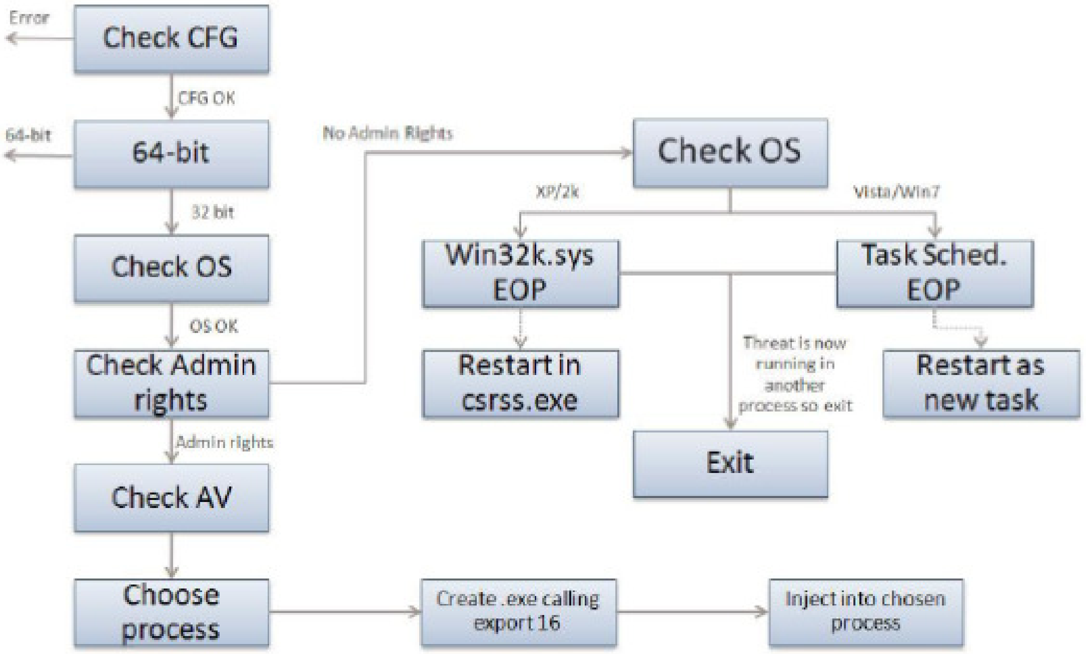


Figure 5-6 How Stuxnet works [21]

5.5.2 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 [22]. 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 uses 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 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 [22].

5.5.3 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 [23].

Detail of the attack

Attackers using several locations in China have leveraged C&C servers on purchased hosted services in the 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

5.5.4 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 USB or click on link on a email that redirect to a fake page). When the employee make 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 computer, 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 know 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) [7].

6 ICS security policies and solutions

In order to better understand how to protect ICS systems, it is important to analyse the security risk of these systems and to understand appropriate security solutions to protect them from attacks.

Most organizations leveraging contemporary ICT are familiar with cyber security issues and what is required to protect critical information assets. Common ICT and ICS security elements include [5]:

- Policies
- Antivirus / antimalware
- Firewalls and Intrusion Detection System (IDS)
- Intrusion Prevention System (IPS)
- Unified Threat management (UTM)

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 present. Contents in a standard cyber security policy can include [5]: Policy upkeep, refinement of policy, and compliance

- Cyber security countermeasures
- Cyber security technologies
- Incident response
- Forensics
- Access control
- Physical security
- Patches and upgrading

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 [5].

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.

6.3 Firewalls and Intrusion Detection Systems

Firewalls are probably the most common security technology found within ICT environments. Most people understand the principle of the firewall and how they provide security [5].

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 [5].

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.

6.3.1 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 gateway of the SCADA network [25].

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.

Among the topologies to enforce separation between the ICT and SCADA networks, [150] 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 6.1) .

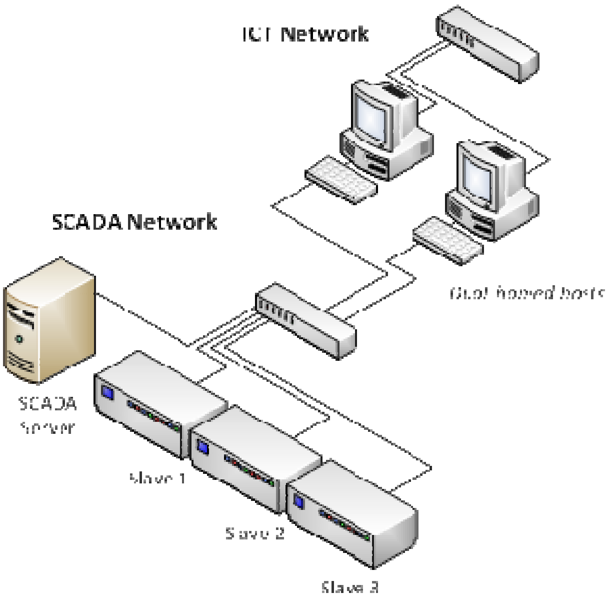


Figure 6-1 Network separation using dual-homed systems

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 [150] found that several organizations were using this topology.

Topologies based in the (two-zone) separation of the ICT and SCADA networks (see Figure 6.2), by means of a firewall/router are a significant improvement in terms of security.

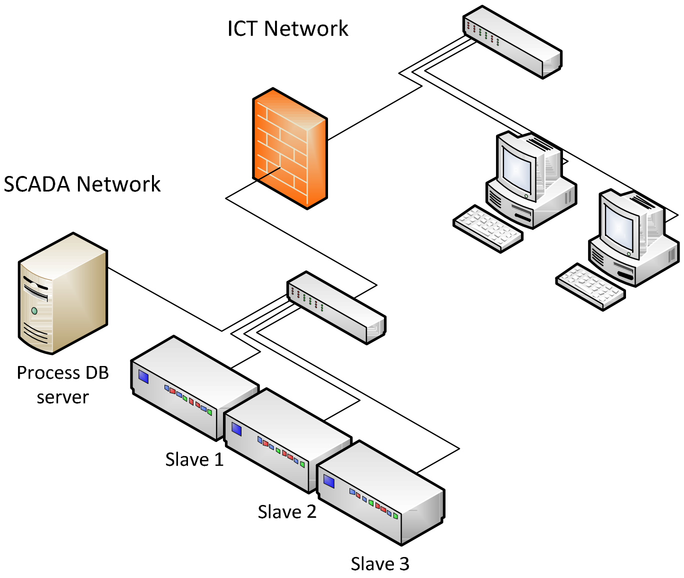


Figure 6-2 ICT and SCADA network separation

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.

The third alternative (see Figure 6.3) is based on a multi-zone topology [151], 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.

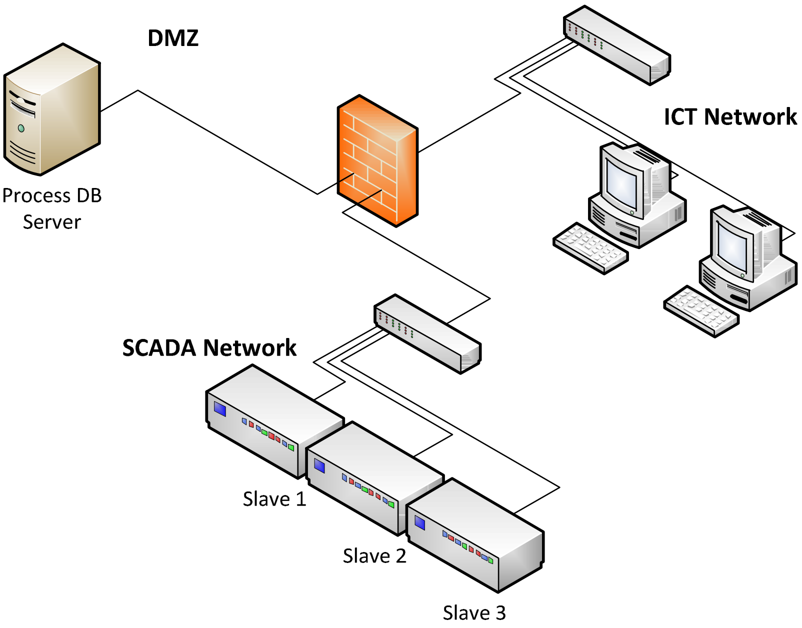


Figure 6-3 Multi-zone topology with a DMZ

In this topology, direct communications between the ICT and SCADA networks can be restricted to a minimum. Role and host-based access control mechanisms (which may also include port-based control based on the 802.1X protocol [152]) might be also added to increase security, in the scope of an AAA (Authentication, Authorization and Accounting) infrastructure designed to provide an added level of security, with non-repudiation and strict control capabilities.

6.3.2 IDS systems

Intrusion Detection Systems (IDSs) provide an add level of security for networks and systems, by providing critical information about attacks. In general, they do not actively block attacks or prevent exploits from being successful, a task that falls into the scope of Intrusion Prevention Systems (IPSs). IDSs fall into two main categories: Host IDS (HIDS), Network IDS (NIDS) and Hybrid IDS (which share the characteristics of NIDS and HIDS).

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 [153] or OSSEC [154] 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 [155] and OSSEC [154] 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.

DARPA's Common Intrusion Detection Framework Architecture (CIDF) [156,157,158,159] 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 [160]), 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 6.4.

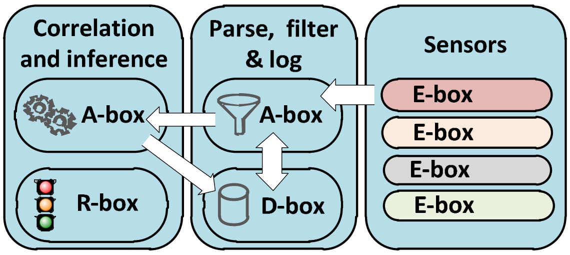


Figure 6-4 CIDF generic IDS architecture

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 [161,162]:

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 SPADE [163] or OSSEC [154]).

2. Anomaly-based detection consists on finding deviant behavior 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) [164,165], Kalman filters [166], PCA (Principal Component Analysis) [167] or Sketches [168,169]. 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 labeled 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 behavioral 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 URCA tool [170], for instance, uses both signature-based and supervised learning techniques. Another example is presented in [171].

   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 [172,173] 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 misjudgment and consequent decision). Botminer [174] 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 [172]). 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, [172] 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 [175]). 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 [153], 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 [176], 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.

6.3.3 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 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 [177]. 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 [177]. 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 6.5 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 [178]. 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.

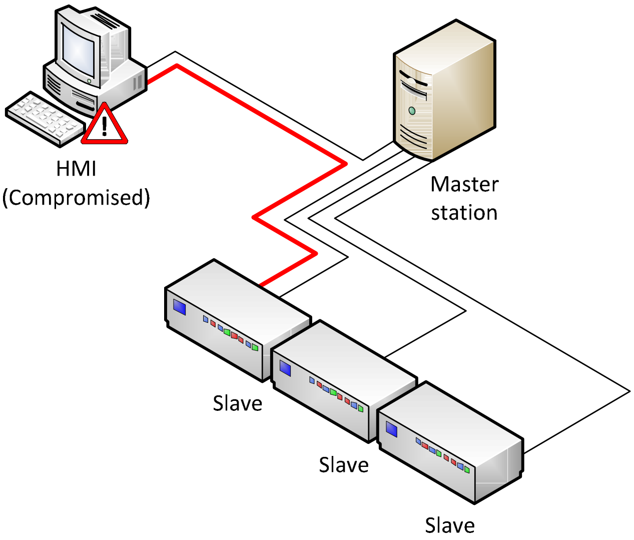


Figure 6-5 Incorrect communication flow in a SCADA system

Figure 6.6 shows an example of a situation where a Master station has become compromised.

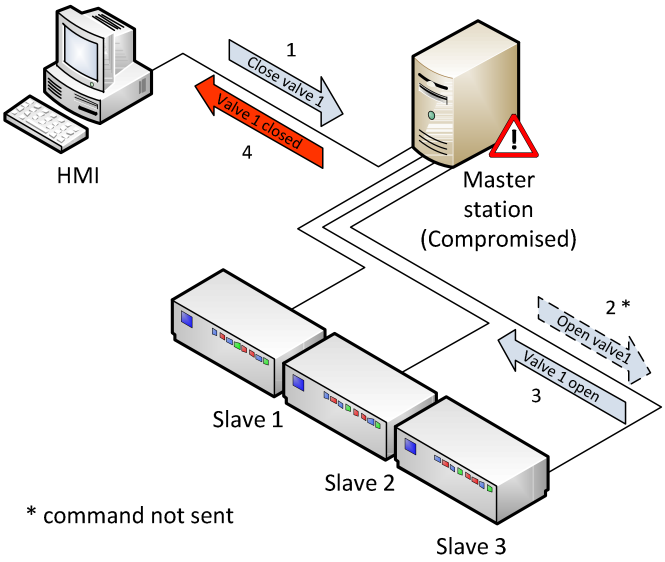


Figure 6-6 Incorrect communication flow in a SCADA system

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.

The Snort IDS [179] has SCADA-specific signature packages, as the ones found on [180], 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 [181], 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 that is 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 analyzing 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 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”

6.3.4 Unified Threat Management

Often, Unified Threat management is referred to by many different names but the essence of the solution remain the same. UTM is about collecting security information from across the ICT architecture and analyzing it at a single location. Operators and security administrators can obtain a common operating picture as it relates to the cyber security of their network. UTM is deployed to include information collected from firewalls, routers, remote access points, wireless access points, IDS, IPS, data flow analysis, and in the security log files, collected from any number of serves in the operational environment [5]. The Automated Systems Management (ASM) framework, from Industrial Defender [182], which is discussed in section 6.4 is one of such tools.

6.3.5 Online Vulnerability Map Tool

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 [183], Metasploit [184], Core Impact [185] and Canvas [186] modular penetration-testing frameworks for which SCADA modules are available (for instance, [187] 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.

Metasploit makes available many exploits and payloads, which can be used for obtaining administrative access to computers, servers and even network equipment. Exploits are used to leverage flaws and/or vulnerabilities found in their systems design, while payloads are the code which can be used to interact with compromised systems (if we make an analogy with a buffer overflow attack with remote code execution, the exploit is the part that allows the attacker to overload the buffer with data to smash the processor stack, deploying a code payload that is to be executed to take control of the system).

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.

6.4 IDS: commercial solutions

Until recently, the IDS commercial product offer for critical control infrastructures was very limited. However, with recent attacks, the rise of new threats and the dissemination of known vulnerabilities on SCADA protocols, the industry has started to develop and offer specific security solutions for these environments, tailored for the particularities of those systems.

6.4.1 A centralized solution: Zenwall

Secure Crossing's Zenwall [188] was designed to filter a wide range of industrial protocols, such as DNP3, Modbus, ProfiNET or OPC (OLE for Process Control) with two standards: OPC classic and OPC UA (Unified Architecture).

It is an appliance-based system which incorporates a Deep Packet Inspection (DPI) firewall, being able log the properties of several connections, including IP addresses, ports involved or the connection and the sequence numbers of the packets traversing the connection. Therefore, Zenwall is able to detect attacks that conventional IDS systems are unable to catch, and stateful firewalls are unable to deter, such as viruses and worms, being also effective against buffer overflow attacks, Denial of Service (DoS) attacks, sophisticated intrusions, and a small percentage of worms that fit within a single packet.

Zenwall uses a scan engine, ZenD, which was built from scratch to filter Industrial Protocols, using FreeBSD for its operating system. The protocol filters inspect all traffic from specific control protocols, based on the standards to which each protocol conforms. As such, ZenD is to analyse the complete data stream, being able to contextually control and filter specific operations on the control infrastructure (such as Reads, Writes, Stops, Resets sent to the

PLCs or RTUs). ZenD is also able to learn from pre-analysed protocol data streams, generating filter templates which can be enhanced by the security operator.

6.4.2 A distributed solution: Industrial defender

Industrial Defender [182] has a portfolio of distributed security solutions, grouped under the Automation Systems Management (ASM) framework (see Figure 6.7),

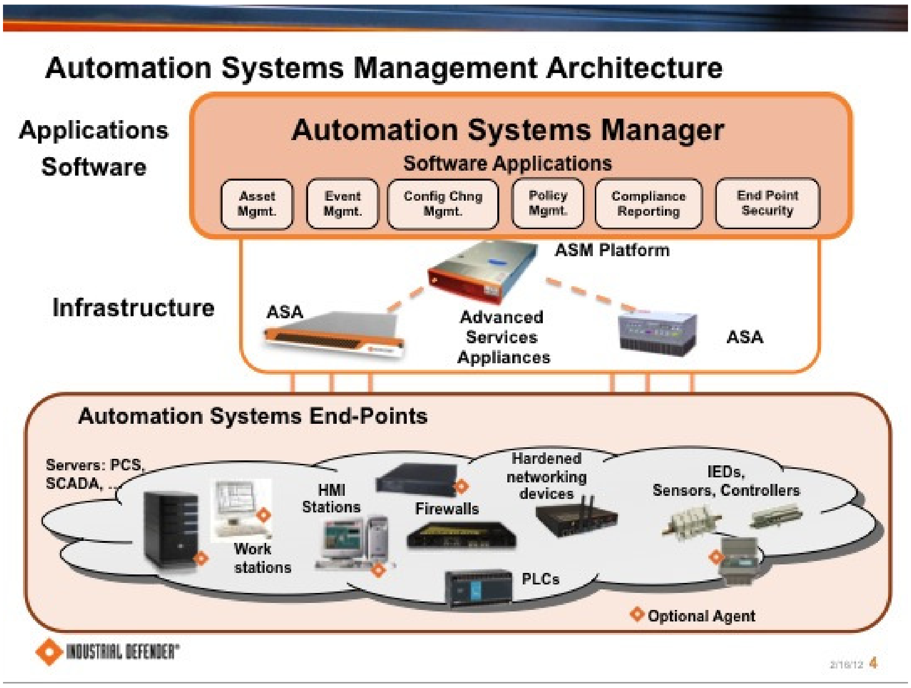


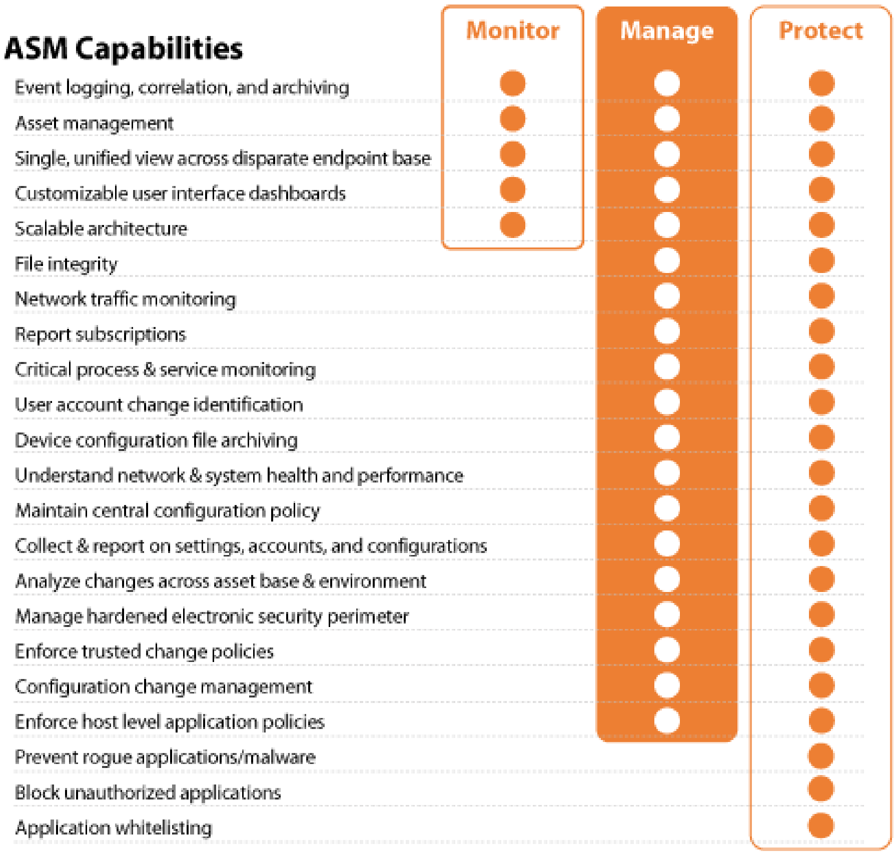
Figure 6-7 Industrial Defender ASM framework [182]

ASM covers the management, monitoring and protection aspects of the SCADA infrastructure. Its portfolio covers the following products (see figure 6.8):

- The *Monitor* solution is a product which covers event processing, correlation and archiving, which can be collected from a heterogeneous array of devices on the control infrastructure. It also provides a centralized dashboard and log data consolidation for forensic purposes.

- The *Manage* solution encompasses all features of the *Monitor* product, adding some Host IDS capabilities (network traffic and event monitoring with signature and integrity checks), management lifecycle features (asset management, change management). The asset management module is able to start with minimal information about the infrastructure, being able to use agents or management APIs like get the remaining information.

- The *Protect* solution encompasses all the features from the *Monitor* and *Manage* products, adding HIDS application and system behaviour monitoring and control features which make use of a whitelisting to restrict execution only to trusted applications, preventing rogue threats and malware dissemination. It also protects against memory tampering with applications and services, integrating with a centralized reporting and administration system.



ASM Capabilities	Monitor	Manage	Protect
Event logging, correlation, and archiving	●	●	●
Asset management	●	●	●
Single, unified view across disparate endpoint base	●	●	●
Customizable user interface dashboards	●	●	●
Scalable architecture	●	●	●
File integrity		●	●
Network traffic monitoring		●	●
Report subscriptions		●	●
Critical process & service monitoring		●	●
User account change identification		●	●
Device configuration file archiving		●	●
Understand network & system health and performance		●	●
Maintain central configuration policy		●	●
Collect & report on settings, accounts, and configurations		●	●
Analyze changes across asset base & environment		●	●
Manage hardened electronic security perimeter		●	●
Enforce trusted change policies		●	●
Configuration change management		●	●
Enforce host level application policies		●	●
Prevent rogue applications/malware			●
Block unauthorized applications			●
Application whitelisting			●

Figure 6-8 Industrial Defender ASM capability portfolio [182]

Accordingly with [189], some of the most distinguishable features of ASM are:

- Asset Management: starting with a minimal set of information, the ASM is able to collect the remainder using both agents and agentless methods, such as WMI (Windows Management Instrumentation) for Windows systems. All the information about system properties (ports, services, software, users) is centralized on the ASM, which enables change monitoring and lifecycle management. An ARP watch feature keeps track of MAC and IP addresses that are not in the ASM to detect unknown devices. Also, alerts can warn when are changes on ports, services or software, on a given system. This module also provides information about existing patches (but does not apply them), being able to import security patch information from vendors.

- Configuration Change Management: configuration change tracking is another feature provided by ASM. Configurations on hosts, services and even network appliances can be tracked, being used to generate alerts. This capability even extends to some kinds of field devices, for instance, to track firmware versions.

- Event Management: the ASM can collect data from a variety of sources, like logs, agents, other security components (such as IDS), being extendable to support other sources of information.

- Dashboard / GUI: in the spirit of Unified Threat Management systems, the ASM provides a highly customizable dashboard that displays all relevant information in an accessible way.

6.5 Secure Communications

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 realtime 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 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 [190] or Arpwatch[191]) 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).

6.6 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 [192]:

- 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 [193] 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 [194] or Honeyd [195] are examples of this honeypot type.

Still regarding honeypot types, there is also another distinction [196] 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 [197], Honeymonkey[198] and CaptureHPC [199].

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 [200] 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.

Gen II honeynets (Figure 6.9) 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 sicne they act as transparent bridges, connecting the Honeynet to the production networks, maintaining the sane address range.

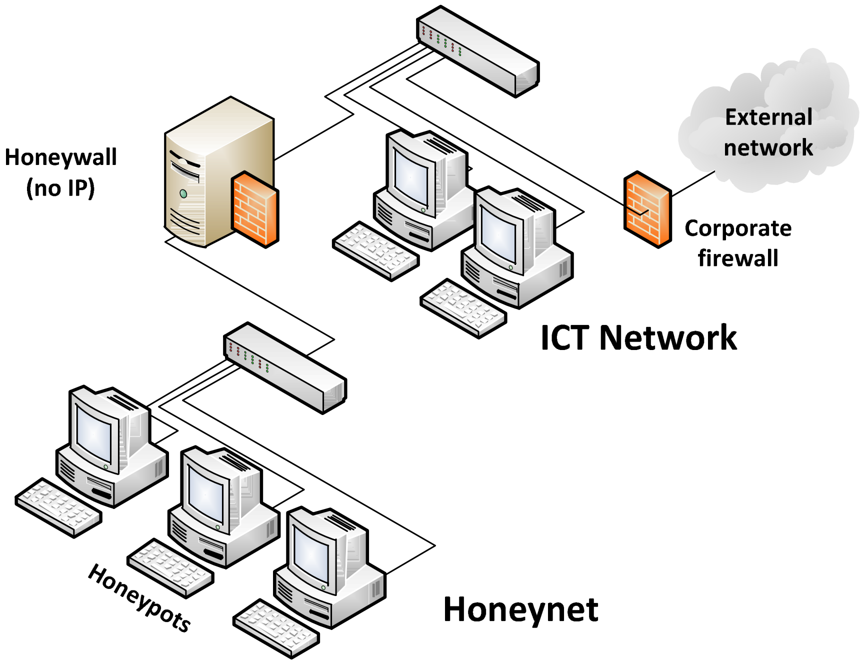


Figure 6-9 Gen II honeynet topology

6.7 ICS security zones

ICS security zones are used to distinguish components, in a large and complex network, at which are required and applied different requirements of security [26]. Particularly, a security zone is a logical grouping of physical, informational, and application assets sharing common security requirements. This concept applies to the electronic environment where some systems are included in the security zone and all others are outside the zone. There can also be zones within zones, or subzones, that provide layered security, giving defense in depth and addressing multiple levels of security requirements. Defense in depth can also be accomplished by assigning different properties to security zones.

A security zone has a border, which is the boundary between included and excluded elements. The concept of a zone also implies the need to access the assets in a zone from both within and without. This defines the communication and access required to allow information and people to move within and between the security zones.

Zones may be considered to be trusted or untrusted.

Security zones can be defined in either a physical sense (a physical zone) or in a logical manner (virtual zone). Physical zones are defined by grouping assets by physical location. In this type of zone it is easy to determine which assets are within each zone. Virtual zones are defined by grouping assets, or parts of physical assets, into security zones based on functionality or other characteristics, rather than the actual location of the assets.

When defining a security zone, an organization must first assess the security goals and then determine whether a particular asset should be considered within the zone or outside the zone. The security goals can be broken down according to the following subsections:

6.7.1 Communications Access

For a group of assets within a security border to provide value, there must be links to assets outside the security zone. This access can be in many forms, including physical movement of assets (products) and people (employees and vendors) or electronic communication with entities outside the security zone. Remote communication is the transfer of information to and from entities that are not in proximity to each other. Remote access is here defined as communication with assets that are outside the perimeter of the security zone being addressed. Local access is usually considered communication between assets within a single security zone.

6.7.2 Physical Access and Proximity

Physical security zones are used to limit access to a particular area because all the systems in that area require the same level of trust of their human operators, maintainers, and developers. This does not preclude having a higher -level physical security zone embedded within a lower-level physical security zone or a higher-level communication access zone within a lower-level physical security zone. For physical zones, locks on doors or other physical means protect against unauthorized access. The boundary is the wall or cabinet that restricts access. Physical zones should have physical boundaries commensurate with the level of security desired, and aligned with other asset security plans .

One example of a physical security zone is a typical manufacturing plant. Authorized people are allowed into the plant by an authorizing agent (security guard or ID), and unauthorized people are restricted from entering by the same authorizing agent and by fences.

Assets that are within the security border are those that must be protected to a given security level, or policy. All devices that are within the border must share the same minimum level of security requirements. In other terms, they must be protected to meet the same security policy.

Protection mechanisms can differ depending on the asset being protected.

Assets that are outside the security zone are by definition at a lesser or different security level. They are not protected to the same security level, and by definition cannot be trusted to the same security level or policy.

6.7.3 Conduits

Information must flow into, out of, and within a security zone. Even in a non-networked system, some communication exists (e.g., intermittent connection of programming devices to create and maintain the systems). To cover the security aspects of communication and to provide a construct to encompass the unique requirements of communications, this standard is defining a special type of security zone: a communications conduit.

A conduit is a particular type of security zone that groups communications that can be logically organized into a grouping of information flows within and also external to a zone. It can be a single service (i.e., a single Ethernet network) or can be made up of multiple data carriers (multiple network cables and direct physical accesses). As with zones, it can be made of both physical and logical constructs. Conduits may connect entities within a zone or may connect different zones.

As with zones, conduits may be either trusted or untrusted. Conduits that do not cross zone boundaries are typically trusted by the communicating processes within the zone. Trusted conduits crossing zone boundaries must use an end-to-end secure process. Untrusted conduits are those that are not at the same level of security as the zone endpoint. In this case the actual communication security becomes the responsibility of the individual channel.

This is illustrated in Figure 6.10.

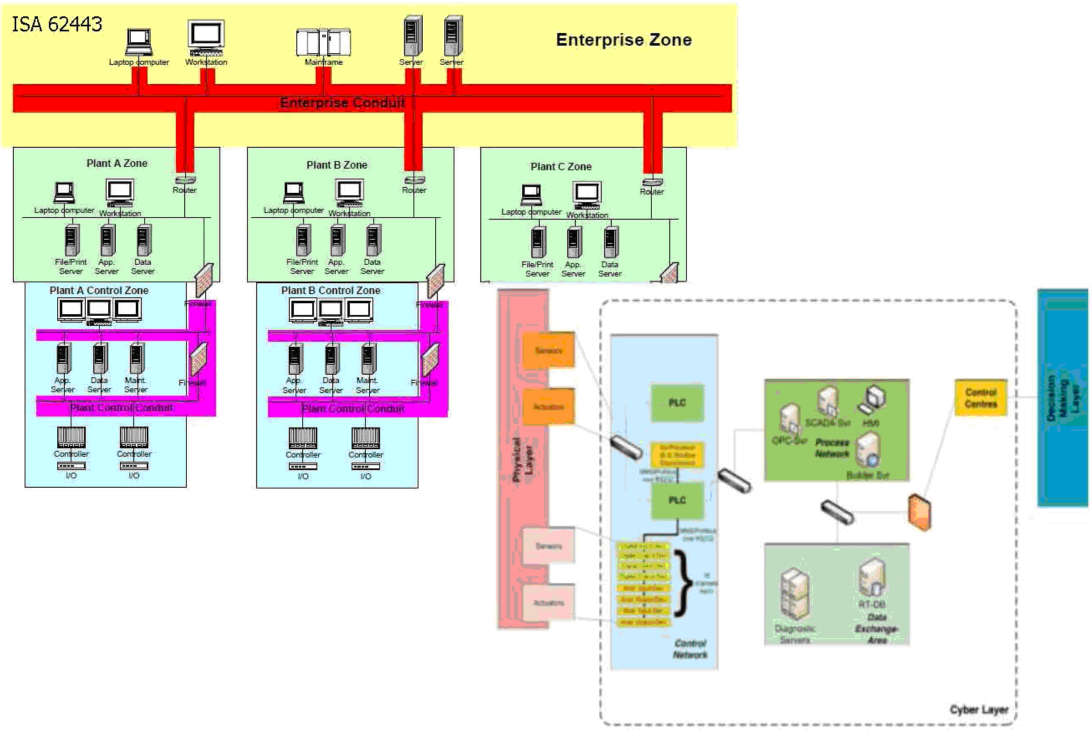


Figure 6-10 Conduits [ISA]

This figure represents a three-plant organization with a separate corporate headquarters. The three plants are connected to the enterprise network to allow communications to headquarters and the other plants. Four possible conduits are defined in the drawing (others would also be defined, but are skipped for brevity). The first is the enterprise conduit, shown at the top of the figure. It connects multiple plants at different locations to the corporate data center. If the wide area network (WAN) is constructed using leased or private communications, then it could be considered a trusted conduit. If it uses both public and private networks, then it may be classified as untrusted. Included in the conduit is all of the communications equipment and firewalls that make up the plant links. Instances of the second conduit class are shown in each plant. Here each of the plants has its own trusted conduit to allow control communication.

6.7.4 Channels

Channels are the specific communication links established within a communication conduit. Channels inherit the security properties of the conduit used as the communication media (i.e., a channel within a secured conduit will maintain the security level of the secured conduit).

Channels may be trusted or untrusted. Trusted channels are communication links that allow secure communication with other security zones. A trusted channel can be used to extend a virtual security zone to include entities outside the physical security zone. Untrusted channels are communication paths that are not at the same level of security as the security zone under study. The communications to and from the reference zone (the zone that defines the communication as non-secure) must be validated before accepting the information.

Figure 6.11 shows an high-level example of systems broken down into zones connected by conduits.

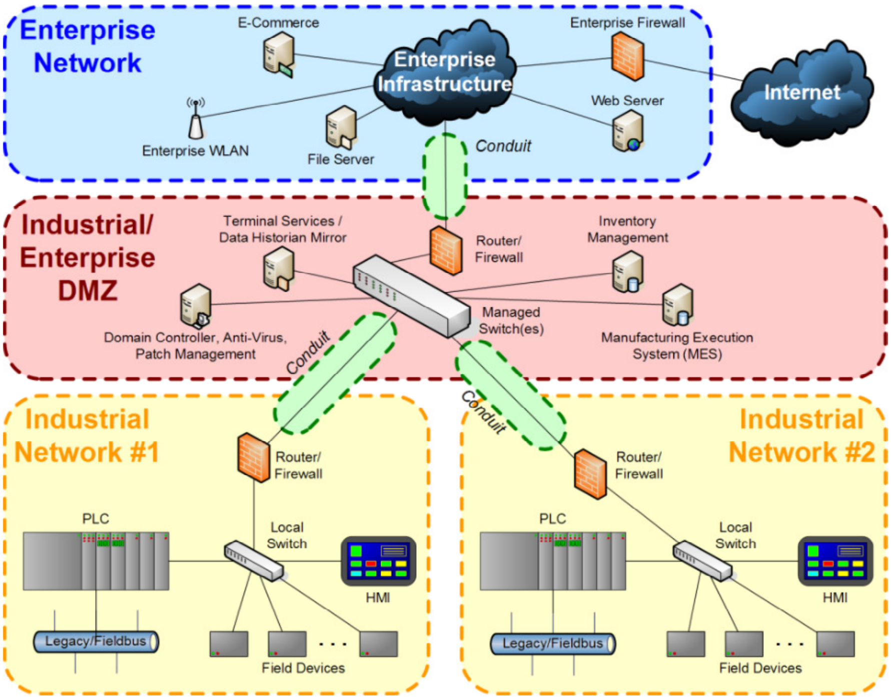


Figure 6-11 High-level manufacturing example showing zones and conduits [ISA99]

6.8 Best Security Practices

Security practices such as computer operation and network management policies must be defined according to guidelines (i.e. NERC [79]) for procedures such as the choice of passwords and their expiry, use of a limited number of privileged computer accounts and disabling the rest, closure of unwanted communication ports and computers, enforcement of access control mechanisms, and frequent update of anti-virus signature databases. It is useful to evaluate the extent to which the corporate and SCADA networks can be logically and physically separated without affecting any functionality, in order to prevent a vulnerability in one network from making the other network also vulnerable.

7 ICS cyber security: modelling techniques and tools

The main objective of CockpitCI project is to demonstrate that tools exist that may reduce the risk of degradation of services delivered to CI customers due to cyber attacks to SCADA systems.

Cyber security methodologies, models and tools are fundamentally based on identification of attacker profiles, attack objectives , attack steps characterization , spreading throughout ICS network and consequences on CI customers.

In this view and having in mind the main objective of CockpitCi project, different cyber security methodologies, models and tools are discussed, used as a single package to address specific aspects of the attack scenario, and/or integrated together to afford the whole attack scenario. In fact, at the state of the art, no single modelling technique has the modelling power and the analytical tractability to adequately deal with the modelling and early prediction of QoS of SCADA system facing adverse events, such as cyber attacks, and accounting cyber interdependency along CI ICT backbone.

As a consequence, for analyzing ICS under cyber attacks and the related consequences on CI (i.e. Power grid) services to customers, we distinguish four kinds of models each one requiring specialized methods and tools which could rely on specialized or not (general) modelling formalisms:

1. Attacks/attacker/vulnerability models
    a. attack/vulnerability trees
    b. Petri nets
    c. Game theory
2. ICS & enterprise network models
    a. network simulators
    b. Emulators
3. CI models
    a. power flow simulators
4. Composite models
    a. to represent more than one aspect of the attack scenario (at least two different kinds of the previous models) till the whole attack scenario (i.e. attacks model plus ICS & enterprise network model plus CI model)
    b. require more than one (Hybrid versus homogeneous) method and tool .

Several tools which cover partially or as whole the above methods and models, are under examination. Some of them are:

PENET, ADVISE, I2SIM,CISIA, NETLOGO, RAO,NS2

According to the underlined formalisms, many of them rely on the stochastic approach as Petri nets, Game theory, Markov chains, Bayesian networks, Monte Carlo methods. Others rely on different approaches such as Agent based simulation , discrete event simulation, etc.

7.1 Stochastic approaches

A stochastic model is a model that involves probabilities, or randomness, associated with time and events. When using such a model, a stochastic process will represent the system behaviour. The stochastic model can be depicted as a state transition diagram, which describes all relevant operational system states and the possible transitions between these states. To describe time aspects between events, a rate matrix has to be specified. One usually assumes that the event that will occur next, as well as the time until the next event, is random. Hence, the behaviour of the system is a stochastic process. The main advantage of this modelling approach is that it captures the dynamic system behaviour, i.e., the sequence and time aspects of events, such as failures and repairs. The stochastic process can then be used as a basis for quantitative analysis of the modelled system. By using mathematical analysis techniques, closed-form solutions may be obtained, which describe how the failure and repair rates affects the expected system dependability in terms of its reliability, availability and so forth. In many cases, the stochastic modelling approach is the most appropriate system evaluation method when quantitative dependability measures are needed.

According to the definition of dependability provided in [27], dependability comprises several system properties, amongst them also the Confidentiality , Integrity, Availability (CIA), typically, security attributes. One would therefore expect that security can be modelled and analyzed by the same methodologies as the other dependability properties. However, it turns out that this is not the case. The main reason is that malicious behaviour is rarely considered as a possible fault source when evaluating system dependability.

This means that the stochastic modelling approach that is so useful when analyzing systems to obtain quantitative measures cannot be applied as it is to evaluate security properties. At the state of the art different approaches try to overcome this problem by proposing methodologies that makes it possible to incorporate attacker behaviour into the transition rates of a stochastic model, so that a comprehensive system evaluation can be performed.

7.1.1 Modelling Malicious Behaviour

Given that a system is represented by a stochastic model, the execution of a transition caused by malicious behaviour will henceforth be referred to as an attack action. It is assumed that a large number of adversaries, i.e., attackers, can target the system simultaneously. This is a realistic assumption for most of the networked ICT systems of today, which are on line round the clock. By studying log files one can see that these systems are constantly subject to more or less suspicious activity, such as probing, worm activity or other kinds of vulnerability exploitation. The rate value of a transition in the stochastic model, which represents an attack action, will then model the accumulated failure intensity, given that all attackers will always try to attack the system. Unfortunately, this rate value is in itself not enough to accurately describe the expected time before the transition actually will occur. One of the main reasons is that attacks are not truly random processes. Because attackers act with intent, they are not always well characterized by models of a random nature [28]. For example, assume that the system that is to be evaluated is a small corporate local area network (LAN) consisting of a private fileserver, a publicly accessible web server and a router connecting the LAN to the Internet. Now assume that the expected time a remote attacker would need to break into and read access restricted files on the fileserver is about the same as the expected time needed to break into and deface the web server. The latter can be characterized as an integrity failure and the former as a confidentiality failure. However, in practice it may be much more common that web servers get defaced than that fileservers get compromised. In fact, the network administrator of this particular LAN assess the frequency of the former to be five times as high as the latter. When using a stochastic model to evaluate this system, the rate values of these two security failures must represent the actual occurrence rates of the events, rather than the success rates of the individual attack actions.

Attacks that are caused by human beings, and that lead to security failures, are very often highly intentional with the specific aim of causing maximum benefit to the adversary or damage to the system. The basic idea is that the probability of an attack will depend on not only the expected time (or effort) required to perform the attack but also on how motivated the particular attacker is. As already seen, there are a number of factors that drive humans to attack computing system, such as financial gain, curiosity, pure entertainment, a rise of ego, etc. On the other hand, a number of factors may reduce the attacker’s motivation and make him refrain from certain attack actions. For example, an employee, with a user account on the corporate LAN , may put his future career at risk if he tries to abuse his insider privileges to attack the local computer network. The gain from a successful break-in into the fileserver may therefore be smaller than the possible consequences he will experience if the intrusion is detected by the system administrator. As another example, the illegal aspect of actions (criminal offense) may prevent even a remote attacker to use available tools to exploit vulnerabilities in such networks. Even though the expected time or effort to perform an attack action may be randomly distributed, the decision to perform the attack will therefore be a trade-off between the gain from a successful attack and the possible consequences of detection.

Attacker behaviour is represented as a probability distribution over all the possible attack actions available in a particular system state. These probabilities are then reflected in the transition rates of the stochastic model by weighting the corresponding (accumulated) attack intensities. For example, if an attacker will choose a particular attack action with probability 0.5, then we can expect 50% of all attackers to take this action, given that they all share the same motivation. Hence, by introducing attack probabilities as parts of the transition rates, the result from a successful attack can be modelled as one or more intentional state changes of the underlying stochastic process, which represents the dynamic behaviour of the system. This is illustrated in figure 7.1 where 1 is a good system state, 2 is a (security) failed system state, a is an attack action, $\lambda_{12}(a)$ is the accumulated attack intensity (given that all attackers always take action a) and $\pi_1(a)$ is the probability of action *a* in state 1.

Some stochastic modelling approaches can be considered high-level approaches in that they focus on the impact of the intrusions on the system rather than on the specific attack procedures themselves. This facilitates the modelling of unknown attacks in terms of generic state transitions. For example, in the stochastic model depicted in figure 7.1 the attack a can simply be explained as “the action that seeks to transfer the system from the good state 1 to the failed state 2”.

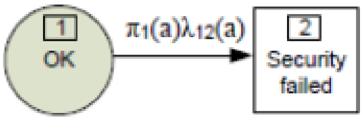


Figure 7-1A stochastic model with assigned failure rate [29].

7.1.2 Attacks modelled as a series of state changes

Attacks on an operating computer system can often be modelled as a series of state changes of the system that lead from an initial secure state to one or more target compromised states, i.e., security breach states. A successful attack against the system may therefore consist of many subsequent elementary attack actions. At each intermediate stage of the attack, the attacker will therefore have the choice of either i) *Attack* by performing the next elementary step in the attack (the system will be transferred from state i to state i + 1, If the attacker succeeds; the system will remain (temporary) in state I, If the attacker fails) or ii) *Resign* and interrupt the ongoing attack (the system will be remain (temporary) in state i). On the other hand, at each intermediate stage, the system administrator may *detect* the attack and bring the system back to a secure state (the system will be transferred from state i to state 0, hence, the attacker will not have the possibility of continuing the attack).

Figure 7.2 shows the attack stages. In the model it is assumed that once an attack is initiated, the attacker will never voluntarily try to revert the system to any of the previous states.

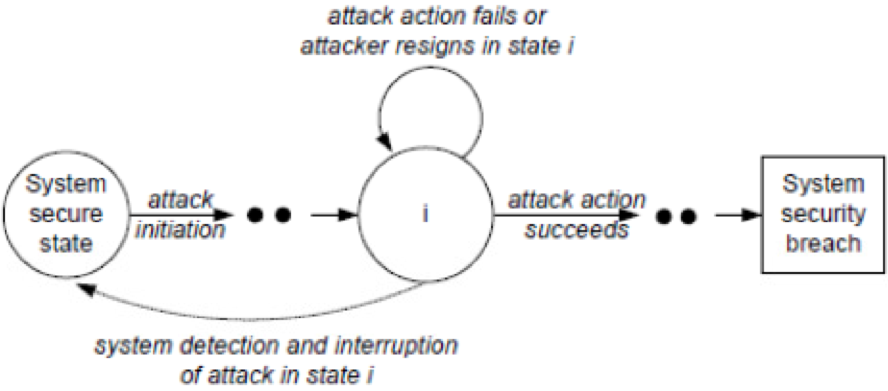

attack action fails or attacker resigns in state i
System secure state
attack initiation
i
attack action succeeds
System security breach
system detection and interruption of attack in state i

Figure 7-2 Penetration of a computer system modelled as a series of state changes [29]

The model also assumes there is only one single path to the security breach state; a somewhat simplified view of reality. Since the state transition model presented in figure 7.2

is stochastic by nature, the time spent in each state of the system model will be a random variable. The time or effort taken for an attacker to cause a transition will depend on several factors, such as the attacker's knowledge and background, robustness of the system etc.

7.2 Game theory

Game theory has been perceived as natural way of modelling cyber security. Indeed, a game is a description of the strategic interaction between opposing, or co-operating, interests where the constraints and payoff for actions are taken into consideration [85]. Depending on the nature and amount of information held y each players locked in a play, a game can be perfect or imperfect, complete or incomplete, static or dynamic.

Perfect/ Imperfect game
A game is labelled perfect when all payers involved in the game are aware of the set of actions that an adversary player has already taken. Conversely, an imperfect game is one where at least one player does not know the next moves of an opponent.

Complete / Incomplete game
A complete game depict one where all the players are well accounted to the strategy of their adversary and their objectives. However, the set of actions that may be taken towards meeting such objectives may not be necessarily known. The distinction between a complete game and a perfect game resides in the fact that it does not take into account the actions each player have already taken [85].
By analogy, a game is said to be incomplete when at least when player is not aware of some of the strategy and objective of a certain layer.

Static game
A game is said to be static if no players can change his/her strategy during the course of the game. Generally speaking, a static game is considered as a one off game as each player plays up his/her strategy in one go without subsequent move left. A static game is an imperfect game by nature as no further information as what the next move of an adversary player will be.

Dynamic game
As opposed to a static game, players in the context of a dynamic game choose their strategies as the game is being unravelled

7.2.1 Game theoretic based approach to cyber-security

[86] have investigated the usefulness of game theory to capture information warfare. In the paper, the authors reviewed four different games before discussing how a dominant position can be achieved and maintained through the orchestration of an appropriate strategy.
The first of such games involves two armies engaged in a military warfare, with one set to use its technological capability to disable the enemy's Command, Control, Communication, and Intelligence (C^3I) before the actual military offensives take place. The second example used by the authors concerns a cyber-attack on such critical infrastructure as nuclear and electricity power plants, telecommunication, water and gas, using DoS tools, virus and others worms. The ultimate aim of the attackers in this case is to wreak havoc and nurture fear, in the midst of the society. The third example discussed by the authors has great similarities to the previous one as it involves a terrorist attack on a number of business and

companies that may be key to the economy of a country. The successful launch of such an attack depends on attacker being able to gather information on the targets and also in determining the optimal timing for such an attack. The fourth and final example involves a dormant warfare which aims at collecting strategic information related to the economy and technology in view of hindering progress.

Having applied a game theory approach to these examples, the authors concluded that:

1. a bold strategy was required to force an enemy to believe that a player will not accept any threats.

2. mixed strategies can mitigate the dominative position of the attacker, especially when any defence strategy is effective only against a specific attack strategy. Changing the defence strategy somehow randomly will increase the probability of mitigating attacks.

3. an attacker should overload a network only part of the time, so that the defender will not stop using a network completely.

4. Maintaining a dominating position requires the stronger player to limit the long term costs to the weaker party since this may otherwise lead to a rebellion leading to damages on both sides.

[84] have argued that a comprehensive grasp of an attacker's intent, objectives and strategies (AIOS) is key to a successful risk assessment and harm prediction. Subsequently, the author proposed a game theoretic approach to inferring AIOS. A brute force DDoS attacks is used as a case study in the experiment conducted by the authors to demonstrate how attack strategies can be inferred in real-world attack defense scenarios. Some of the key findings of the authors are that the security and assurance of the system greatly depends on the appropriate selection of the game model. Furthermore, the effectiveness of the IDS and the correlation of the attack actions play a role in the determination of the best AIOS game models.

[85] adopt a game theoretical approach to the modelling of a DoS and DDoS in network systems. The precept of such initiative lies on the potential of game theory concepts to capture the realm of cyber security: that of two entities competing for contradictory pay offs. Indeed, a distributed denial of service is modelled as a two-player game in which the attacker attempts to find the most effective packet sending rate or botnet size, while the defender or network administrator is concerned with putting in place the best firewall setting to block unwanted traffics while allowing the legitimate traffics through. A DoS is represented with a single attacking node while multiple nodes are used in the context of a DDoS. In both cases, the authors assumed that the malicious nodes are operated by one attacker and that, two possible cases can be considered. The first of such cases considers the game as being static i.e. neither the administrator nor the attacker change their strategy during the course of the game. In this set up, the strategy of the attacker is confined to a couple of actions including: the selection of the malicious nodes, the sized of the botnet (m) to launch the (D)DoS and the set-up of the rate of malicious traffic (r_A). Conversely, the defender or network administrator can only change the mid-point (M) of the firewall which represents the rate of packet being dropped by the firewall.
The Nash equilibrium of this game is defined to be a pair of strategies (r_A m, M), which represent the best strategy for both players. The author remarked that the peculiarity of a

dynamic game makes it hard to actually compute the Nash equilibrium since the change in strategy by both players may result in a continuous shift of the latter. For instance, the authors highlighted that, an attacker A can think that if he/she sets r_A low and m high during the first few time steps, the defender D will set M to a low value, and then A can exploit it by setting r_A high and m low in the next few time steps assuming that D does not change M.
A similar reasoning can be adopted by the defender based on assumption made about the attacker' behaviour.

A Markov game approach to the assessment of risks is proposed by [83]. The authors argued that a comprehensive assessment of risk in network information systems should account of, not only the current, but also the future risks. The work of Xiaolin et al.(2008) is based on the extension of the relationship between threat, vulnerability and asset commonly used in the determination of a risk level. They noted that a vulnerability that remains unpatched can help in the spread of risk, while a risk can be considerably reduced if a prompt and decisive action is taken by the administrator. Subsequently, [83] proposed a game of where the threat and the vulnerability agents are represented as the players. Thus the threat agent increases the risk by through the action "threat spreading" and the vulnerability agent decreases the risk by through the action "system administrator's repairing the vulnerability". The ultimate aim of the game is to a get more comprehensive value of risk as well as giving enabling the system administrator to select the best system repair scheme.

7.3 Attack Trees

Attack trees were introduced by Schneier [36] as a way of formally analyzing the security of systems and subsystems based on varying attacks. This is basically FTA with the attack goal in place of a fault and basic event probabilities are not failure rates. Schneier's work is notable because it was the first to apply this approach to the area of information security. The attack goal is the root of the tree and the different ways of accomplishing the attack are the leaves, with connections via AND and OR nodes.

Moore et al, [37] describe and illustrate an approach for documenting attacks on software systems using attack tree information in a structured and reusable form. Analysts can then use the approach to document and identify commonly occurring attack patterns and then modify attack trees to enhance security development.

Most recently, attack trees have been applied to a SCADA communication system [38]. The authors identified eleven attacker goals and associated security vulnerabilities in the specifications and development of typical SCADA systems. The team defined eleven such goals:

1. Gain SCADA System Access
2. Identify MODBUS Device
3. Disrupt Master-Slave Communications
4. Disable Slave
5. Read Data from Slave

6. Write Data to Slave
7. Program Slave
8. Compromise Slave
9. Disable Master
10. Write Data to Master
11. Compromise Master

Each goal was ranked roughly in terms of the potential severity of impact (e.g. reading data from a slave device is likely less serious as compared to writing data to the slave). Figure 7.3 shows these basic relationships and ranking. In addition, the study team defined four Supporting Goals that would likely not be an end goal on their own, but would be often required by an attacker to achieve his or her objectives. Each is used in more than one attacker goal. These include:

- Denial of Service Against Networked Device
- Intercept or Modify Data Through Man-in-the-Middle (MITM) Attack
- TCP Sequence Number Attack
- Sniff Traffic

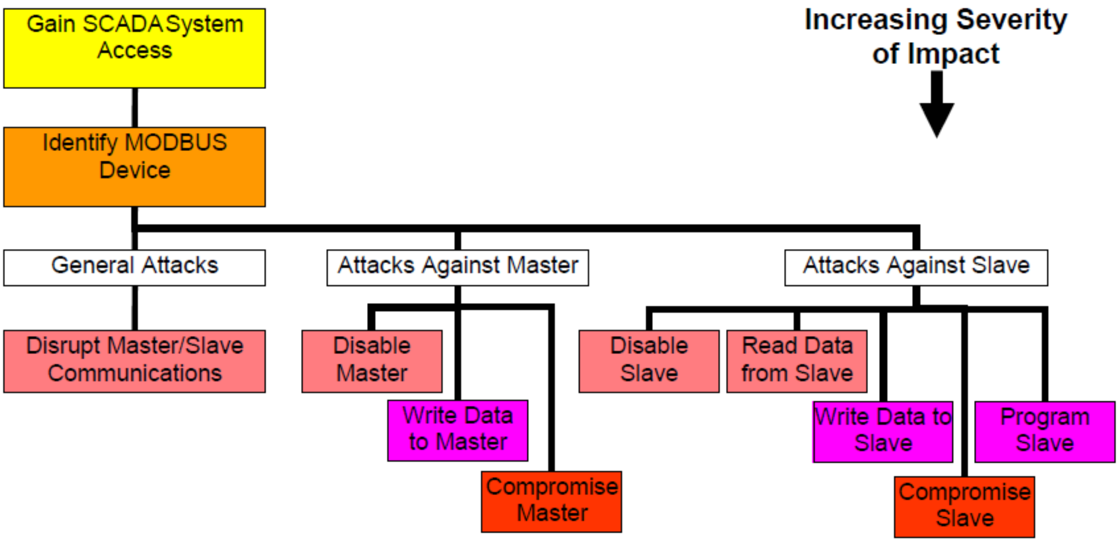


Figure 7-3 Interrelations and approximate severity of attacker goals [37]

On such basis they suggested best practices for SCADA operators and improvements to the MODBUS standard. Their application was qualitative in that attack tree analysis was used only to identify paths and qualify the severity of impact, probability of detection, and level of difficulty. They did not calculate the probability of an actual attack being successful.

A related approach that arose in the computer and information security literature is vulnerability tree analysis. Vulnerability trees are hierarchy trees constructed as a result of the relationship between one vulnerability and another vulnerability and or steps that a threat agent has to carry out to reach the top of the tree [39]. Vulnerability trees help security analysts understand and analyze different attack scenarios that a threat agent might follow to exploit a vulnerability. With this understanding, countermeasures can be taken. The top of the tree is known as the top vulnerability or the parent vulnerability. There are a large number of ways that such a top vulnerability can be exploited. Each of these ways will constitute a branch of the tree. The branches will be constructed by child vulnerabilities. Consequently the child vulnerabilities can be exploited by steps that the threat agent will have to perform in order to get to the parent. Each vulnerability will have to be broken down in a similar way. Normally this will end up in more than one levels of decomposition. When the point is reached where the branches contain only steps, and no child vulnerabilities, then we know that we have reach the lowest level of decomposition (the “step-only” level).

7.3.1 Gain SCADA access attack tree

Attack tree can be seen as a multi-level hierarchical structure based on logical AND and OR operators.

The top node is the ultimate goal with the grouping of different sub goals. The grouping can be composed with a number of attack leaves that are attributed with logic operators “AND” or “OR”. To build an attack tree also vulnerabilities of the system under attack have to be exploited. In [1], three vulnerability indices are introduced: system, scenario, and leaf vulnerabilities, accounting the power system control framework based on existing cyber security conditions.

To evaluate the vulnerability indices in a systematic manner, the following steps are followed:

1. Identify adversary attack objectives.

2. Identify possible security vulnerability and construct the attack tree.

3. Determine the combination of intrusion scenarios with each cyber security condition on each attack leaf.

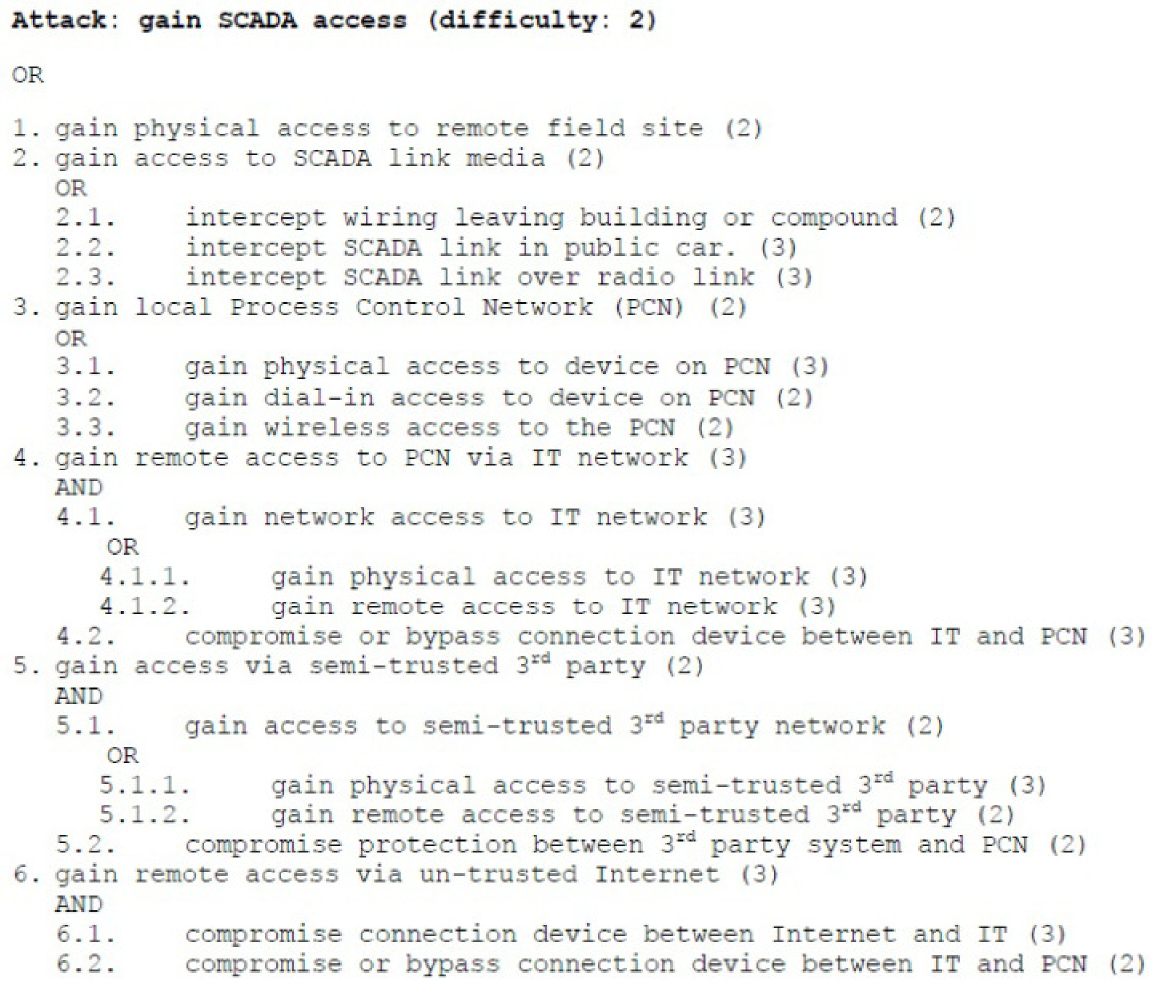
4. Compute leaf vulnerability with respect to the password enforcement and existing technological implementations, given that the cyber security conditions are determined.

5. Scenario vulnerability can be computed according to the combination of corresponding leaf vulnerability indices.

6. Finally, determine the pivotal attack, i.e., system vulnerability based on scenarios’ vulnerabilities, and improve system security.

Figure: 7.4 illustrates a possible attack tree for a SCADA system [13], where the difficulty to reach each point can be estimated. To reach the main goal (the one in bold) is needed just to reach one of the sub goals (one of the possible choices from 1 to 6). Such sub-goals have different ways to be reached, and all work in recursive way. For each leaf , a label can be set to indicate the difficulty of reaching the related sub goal. For setting the difficulty grade of their father, if the sons are in AND, the difficulty grade of the father is chosen as the max of the difficulty grade of his sons, while in case the sons are in OR , the difficulty grade of the father is chosen as the minimum of the difficulty grade of his sons.

The attack trees are very scalable because different trees can be easily joined to make a bigger tree. The limit of the attack tree approach is twofold , one is that enough knowledge is needed to go in deep details as possible, second one is that the attack tree remains on the paper, but can be supported by automatic tools. The natural evolution is towards the use of Petri nets attack models. Petri nets introduce the concept of attack restoration but reduce the scalability of the related models.



```
Attack: gain SCADA access (difficulty: 2)

OR

1. gain physical access to remote field site (2)
2. gain access to SCADA link media (2)
   OR
   2.1.    intercept wiring leaving building or compound (2)
   2.2.    intercept SCADA link in public car. (3)
   2.3.    intercept SCADA link over radio link (3)
3. gain local Process Control Network (PCN) (2)
   OR
   3.1.    gain physical access to device on PCN (3)
   3.2.    gain dial-in access to device on PCN (2)
   3.3.    gain wireless access to the PCN (2)
4. gain remote access to PCN via IT network (3)
   AND
   4.1.    gain network access to IT network (3)
     OR
     4.1.1.     gain physical access to IT network (3)
     4.1.2.     gain remote access to IT network (3)
   4.2.    compromise or bypass connection device between IT and PCN (3)
5. gain access via semi-trusted 3rd party (2)
   AND
   5.1.    gain access to semi-trusted 3rd party network (2)
     OR
     5.1.1.     gain physical access to semi-trusted 3rd party (3)
     5.1.2.     gain remote access to semi-trusted 3rd party (2)
   5.2.    compromise protection between 3rd party system and PCN (2)
6. gain remote access via un-trusted Internet (3)
   AND
   6.1.    compromise connection device between Internet and IT (3)
   6.2.    compromise or bypass connection device between IT and PCN (2)
```

Figure 7-4 Attack goal: gain SCADA access

7.3.2 Attack tree tools

Several tools are available even on the commercial site to implement attack trees. A short description of the attack tree provided by *Isograph*, named AttackTree+, follows.

AttackTree+, through the use of attack tree models, allows the user to model the probability that different attacks will succeed. AttackTree+ also allows users to define indicators that quantify the cost of an attack, the operational difficulty in mounting the attack and any other relevant quantifiable measure that may be of interest.

Questions such as 'which attacks have the highest probability of success at a low cost to the attacker?' or 'which attacks have the highest probability of success with no special equipment required?' can be answered using AttackTree+.

In AttackTree+, different categories and levels of consequence may also be assigned to nodes in the attack tree. A successful attack may have financial, political, operational and safety consequences. A partially successful attack may have a different level of consequence to a totally successful attack. All these types of consequence measure may be modeled in AttackTree+.

7.4 Petri nets

Petri nets (PN) [30,31,32], in their various shapes and sizes, have been used for the study of the qualitative properties of systems exhibiting concurrency and synchronization characteristics.

The use of PN-based techniques for the quantitative analysis of systems requires the introduction of temporal specifications in the basic, untimed models.

This fact has been recognized since a fairly long time, and several different proposals for the introduction of temporal specifications in PN have appeared in the literature. The main alternatives that characterize the different proposals concern

- the PN elements (either places or transitions) with which .timing is associated,
- the semantics of the firing in the case of timed transitions (either atomic firing or firing in three phases),
- the nature of the temporal specification (either deterministic or probabilistic).

In [33] the idea of using PN for attack analysis introduced by McDermott in [12di 33] and extended by others such as Zhou et al. [13 di 33] to add some advantages (Colored PN (CPNs), mapping an attack tree to a CPN) or Dahl [14 di 33] (concurrency and attack model) is followed too.

Particularly, they add a new algorithm for the automatic generation of Petri nets from the description of a SCADA network and its vulnerabilities and propose an approach for risk measures that account for any reachable attack state, given initial conditions on the attacker's access to network resources and host configuration on the network.

Using these measures, ones can explicitly account for all high-consequence attack states, irrespective of likelihood, and support a more flexible notion of risk that can be resolved as one of several computable measures on the discrete attack space. The techniques for evaluating these metrics are based on a Petri net's minimal cover ability set.

7.4.1 A SCADA network attack model

Considering operations against target networks characterized in terms of attack steps, where each step accomplishes one or more of the following [33]:

1. improved knowledge of the target network through reconnaissance,

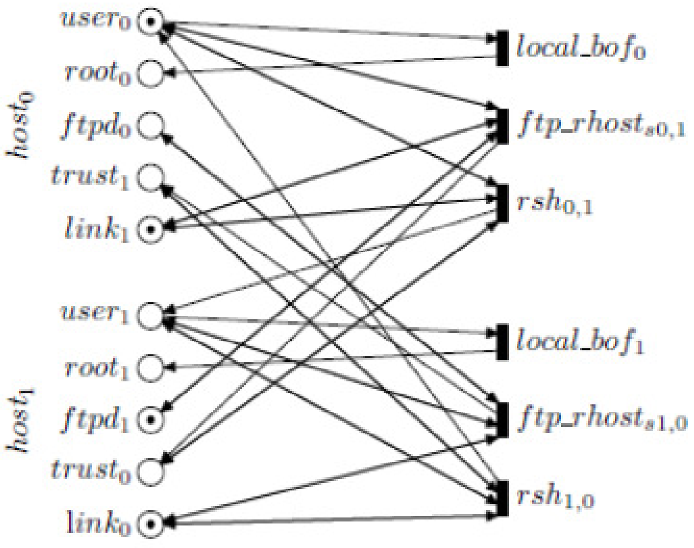
2. access to one or more hosts on the network through exploitation of a software vulnerability or the deception of a legitimate user,

3. increased privilege on one or more hosts on the network through exploitation of a software vulnerability or the deception of a legitimate user,

4. the establishment of sustainable access to one or more hosts on the network by, for example installing a back door, or

5. viewing, stealing, manipulating, or preventing legitimate access to protected information

a PN model for a network attack scenario is displayed in Figure 7.4. In the model, each attack step is represented by a transition, arrows that point in from places represent preconditions, and arrows that point out to places represent post conditions. The places in the PN of Figure 7.5 represent host attributes in the network being modelled. The attributes and associated places in Figure 7.5 include privilege levels (useri, rooti), services (ftpdi), trust relationships (trusti), and connectivity (linki).



Figure

Figure 7-5 Example attack net [33]

Ph is the set of places corresponding to host *h*. In order to represent the fact that *h* is characterized by a particular attribute, the corresponding place is marked by a token. Thus *Ph* represents the attributes that host *h* can have; the places in *Ph* that are marked represent the attributes that *h* actually does have. For example, the place *ftpd1* ∈ *Ph1* is marked by a token, indicating that *host1* is running an ftp server, while the place *ftpd0* ∈ *Ph0* is not marked, indicating that *host0* is not running an ftp server.

For the purposes of attack analysis, transitions represent exploits of vulnerabilities such as buffer overflow (local *bofi*), ftp (ftp *rhosti,j*), and rsh (*rshi,j*). An exploit is intended as any action an attacker takes, including what ordinarily would count as legitimate use of resources, such as the use of rsh. For every exploit *e* there is a set of preconditions, represented by a set of places *pre(e*); and a set of post conditions, represented by set of places *post(e).* In the example, a precondition for performing a local buffer overflow exploit is that the attacker has user access on the target host, and a post condition is that the attacker has root access on the target host. Therefore, for each *host hi*, *useri* ∈ *pre(local bofi),* and *rooti* ∈ *post(local bofi).* The actual occurrence of an exploit is represented by the firing of the corresponding transition. An algorithm has been used to auto-generate the attack Petri net, that executes in three phases: an initialization phase and two processing phases. The initial marking *m0* of the net indicates the conditions that have been met before any transitions in T have fired.

SCADA network on which the attack Petri Net model has been built is comprised of a data historian, a human-machine interface (HMI), an engineering workstation, a master terminal unit (MTU), three remote terminal units (RTU), and two programmable logic controllers (PLC), as shown in Figure 7.6. The MTU communicates with the RTUs and IEDs via a radio serial link (RSL), the maintenance server is accessible via dial-up modem from the public switched telephone network (PSTN), and all other communication is conducted over TCP/IP on Ethernet. In one modelled configuration, a firewall (FW) is used to control traffic between the SCADA network, corporate network (LAN), and the maintenance network. In alternate configurations the historian and workstations are also isolated by the firewall. That is, they reside in separate so-called “demilitarized zones” (DMZs).

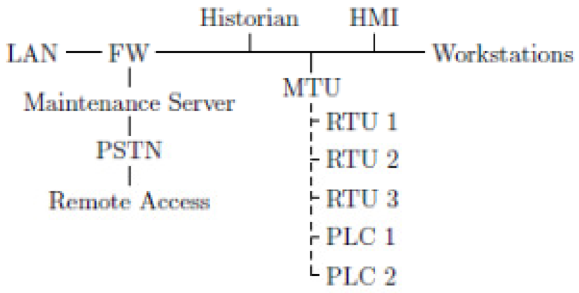


Figure 7-6 Sample SCADA network [33]

Figure 7.7 illustrates the PN model of remote manual operation of a valve.

To open the valve, an operator must issue an open command at the HMI, and the valve’s state at the HMI must closed. If these preconditions are met, the HMI relays the command to the MTU via the Ethernet connection, the MTU communicates the command to the appropriate RTU via the RSL, the RTU driver delivers power to actuate the valve, and the open state is then registered at the RTU and relayed back to the HMI through the MTU.

Of the large number of possible process failures, [33] discusses six in detail by describing the corresponding component failure, the state of the process at the time of failure, and the resulting impact.

Each process failure is related to a set of SCADA attacks, where each SCADA attack has the same result as the induced process failure, but is caused by an attack on the SCADA computing infrastructure. Moreover, for each process failure, the authors assign a measure of its severity in terms of expected number of personnel injuries due to inhalation or skin irritation, by ammonia. They relate this process failure and associated consequence to a set of attacks on the SCADA system as shown in Figure 7.8.

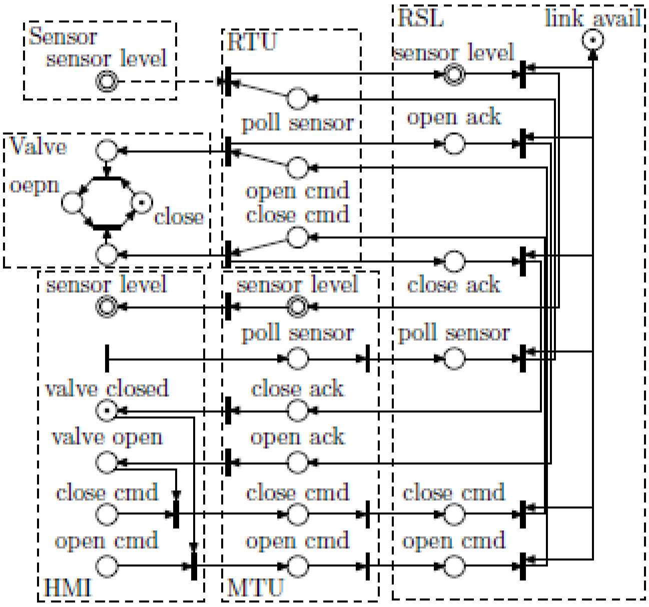


Figure 7-7 Remote Manual Operation [33]

In failure mode *(FM) 1.1* the attacker gains user privileges on the HMI and issues a command to open the valve *v11* before the execution of Task 4, and ammonia will discharge into the dilution drum.

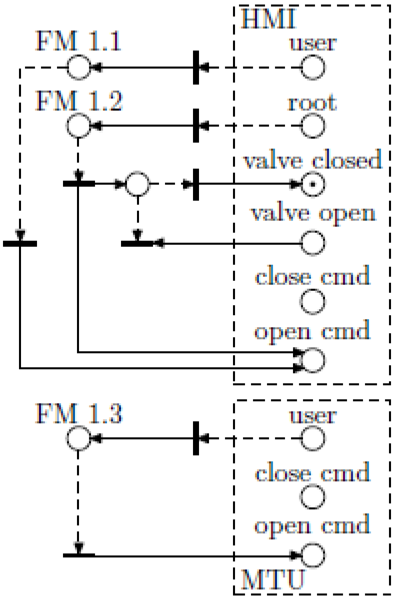


Figure 7-8 Attack-induced Process Failures [33]

A similar, but possibly more devastating attack can occur in *FM 1.2* when and attacker gains root privileges on the HMI, opens valve *v11* before Task4, and spoofs a closed state for *v11*. This attack gives the legitimate HMI operator the impression that the process state is correct for the task at hand and can increase the amount of ammonia discharged. As a result, the expectation of injuries doubles. A third attack (*FM 1.3*) targets the MTU. This attack has the same effects as the HMI super-user attack.

Using coverability analysis, they can determine all of the resources an attacker can acquire in the SCADA network. The SCADA attack set will map those sets of resources to SCADA failure modes that can be induced by the attacker, and the system model will analyze the impact of that failure mode.

7.4.2 PENET tool

Among tools based on Petri nets , PENET tool introduces concepts such as the dynamic nature of attacks [5, 34], the reparability of a system, and the existence of reoccurring attacks.

It attempts to find a balance between ease of use and representation power by providing a set of constructs, parameters, performance metrics, and a time domain analysis of attacks. Particularly, users can draw model diagrams of a given system throughout an intuitive user interface, perform time-domain simulations and carry out security evaluations.

Time-domain analysis produces outputs such as “time to reach the main goal” and the “path taken” by the attacker.

PENET Tool was completely written in C# .NET using Visual Studio 2005 as a development environment. It requires .NET 2.0 framework to run. Because of these requirements, it is not suitable for operating systems other than Microsoft Windows.

The main contribution of the tool is to extend modelling capabilities of attack trees by using Petri net constructs in order to significantly improve the analytical capabilities of attack trees, specifically by:

- Addressing existing issues in attack trees such as limited representation power, imprecision, and lack of defined defense modelling.
- Introducing concepts of recurring attacks, defense modelling, and dynamic constructs.
- Introducing an analysis approach that follows attack execution in time domain.
- Providing means to evaluate system survivability and defense strategies.

Primary audience of this tool is individuals and organizations who want to use such a tool in vulnerability evaluation of cyber attacks and developing defense strategies for their systems. Secondary audience is research community desiring to learn more about attacker behaviour modelling and PENET approach. By the way, it's not available at the moment (22/12/2011) [34].

7.4.3 Stochastic Petri Net Package

Since attacks occur randomly, a stochastic process can be used for the model. In some studies, [59] ,the intrusion and cyber-net are modelled by a generalized stochastic Petri net (GSPN) model [60,35]. The states of the stochastic process are the status of intrusions to a network that are inferred from the abnormal activities. These include malicious packets flowing through pre-defined firewall rules and failed logon password on the computer system. Transition probabilities are obtained from the abnormal activity data in the system.

A GSPN consists of two different transition classes: immediate and timed transitions. As depicted in figure 7.9, which is an illustration of a firewall model that will be discussed later, a status node is represented by a circle. An arrow head denotes a transition of the system status. An immediate transition is shown as a solid bar. Immediate transitions are assigned probability values. Timed transitions denoted by empty bars have delay times associated with the response that an attacker receives from the system. Tokens (dots inside a circle) are used to model the number of intrusion attempts where an attack starts. Token passing describes the change of each transition, or marking.

SCADA systems typically have specially designed firewall rules and password policies to achieve a high level of computer security. A firewall is a technology of cyber security defense that regulates the packets flowing between two networks. As there may be different security trust levels between networks, a set of firewall rules is configured to filter out unnecessary traffic. These rules are written with the following criteria for acceptance or rejection:

1) Type of protocols

2) Incoming and outgoing traffic

3) Specific port service or a port service range

4) Specific IP address or an IP address range

These audit fields are recorded in a firewall and are used offline by a system administrator to analyze malicious behaviours. Due to the high volume of daily network traffic, it is not practical for a system administrator to monitor the network with the available datasets. Thus, an add-on commercial firewall analyzer is implemented to detect anomalies in these datasets.

The malicious packets flowing through a firewall must be identified. Together with the traffic denied by the firewall, such data can determine the probability of cyber attack occurrences either being granted access or being attempted. These datasets can be analyzed from the firewall logs in two ways:

1) The number of records rejected compared to the total number of firewall traffic records, and

2) The number of malicious records bypassing compared with total records for each rule.

The firewall model depicted in figure 7. 9 includes n paths corresponding to n rules in the firewall model. The attacker receives responses from the system through the feedback paths starting with the circles representing rules. The paths vertically passing the circles representing rules are successful attempts.

This model consists of two terminals that can be connected to other submodels. For instance, a network that consists of three zones, including a demilitarized zone (DMZ), can be modeled by connecting two firewall models in series. The construction of the model conforms to the number of rules that are implemented in the firewall. In case the number of firewall rules is large, only a subset of rules considered potentially malicious are included in the formulation.

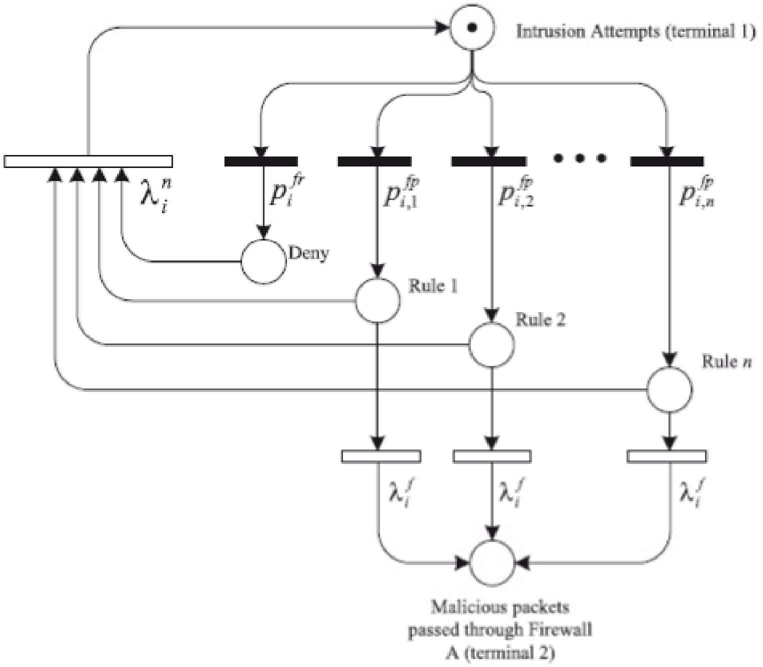


Figure 7-9 Firewall Model with Malicious n Rules [60]

The submodel consists of circles that are the states representing the denial or access of each rule. Each solid bar is assigned a firewall penetration probability that can be calculated from firewall logs.

7.5 SIR Model of Epidemics

SIR is an epidemics based model [61] that may be used in cyber security to study how a malware infection spread among different machines. SIR stands for Susceptible, Infected , Recovered. SIR model represents a disease spread where individuals are susceptible to a disease, potentially contract the disease, recover and become immune to future infections after recovery. There is also a variant of SIR called Susceptible, Infected, Removed, that allow infected individuals to die due to the disease and thus leave the considered population. An individual potentially moves from the susceptible to the infected group when s/he comes in contact with an infected individual.

Given specific assumptions on the average number of spread transmission possible from a given infected individual in each period and on the recovering rate of each individual, there are specific algorithms that shows the result of spread transmission; in [61] if individuals are going to die from an infectious disease it is better that they die fast for the purpose of ending the epidemic; the other result is that it is not needed to immunize everyone in the population in order to prevent an epidemic.

There are several analogies between the malware and the epidemics affecting the animal word. Cyber security domain considers each individual as a machine that may be infected by a malware and a recover capability as the action of antivirus software that are in place to remove the infections. Dying individuals represent the machines that have been fatally compromised. In [61], an individual can pass from S to I and from I to R. When R is reached, the subject is removed from the study (this can occur for death or because the subject become immune to this disease). The passage between each state is governed by several variables. In [62,63], the work of Tassier [61] has been tailored to deal with cyber disease spreading along an ICT network composed by a SCADA system interconnected to a corporate network. The network has been simply described by a graph. Each ICT device (an individual in [61]) is a node of the network, and there is an arc if two nodes can communicate each other (the arcs are symmetric).

The Susceptible, Infected and Resistant (SIR) model was originally developed to study the evolution of a disease over a population, where each individual could be susceptible to the infection, having contracted the infection, or be immune/resistant. The similarity with malwares is very high but ICT models as the one in [62,63] got the problem that model variables have different values depending on the cyber security solution adopted for each kind of node.

Let $\mathbf{N}$ be the number of the nodes of the net, it is constant. Let $j = 1, \ldots, N$ each node, and for every node, d_j is the number of the neighbours of the node $\mathbf{j}$. α is the malware spread value. There are several kinds of malwares, and we can differentiate them for the spreading velocity. A malware that spread itself too fast can be easily detectable due to the high traffic on the net. So, $\beta_j = \alpha \cdot d_j$ indicates on how many neighbours the malware send itself. Malware spread itself just on β_j nodes.

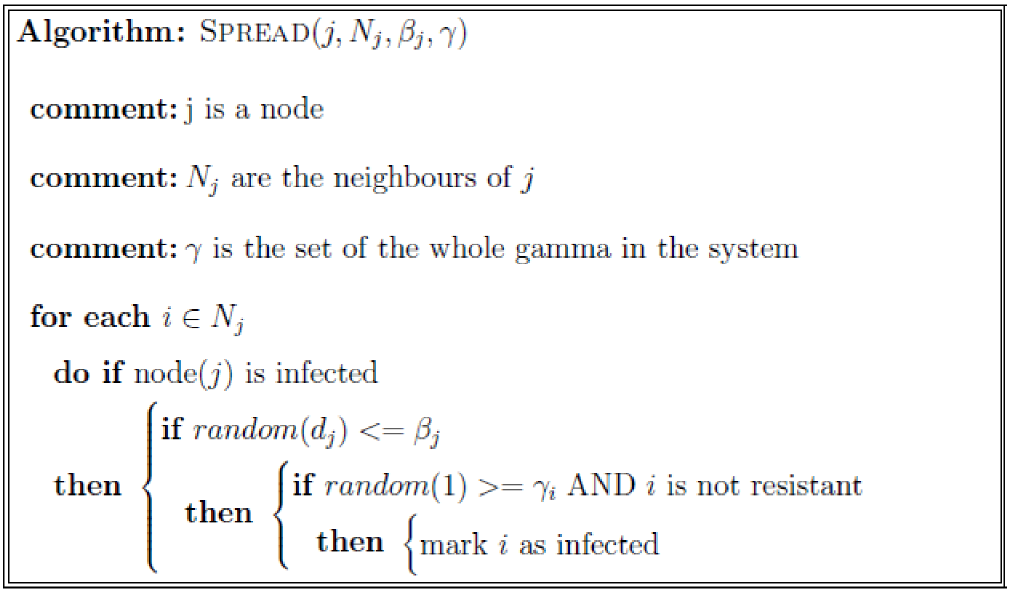
Each host device gets its own security policies (e.g. system patched), or simply relies on an operative system non compatible with the specific malware (e.g. a malware written for Windows cannot infect a Linux machine). A system full patched isn't secure 100% because there are always the zero-days vulnerabilities. Let γ_j that probability (probability to contract the malware).

In [61], once a node becomes infected, a variable ($\mathbf{k}$) keeps into account that after a certain time the node automatically will become resistant. In [62,63] to remove the malware it's necessary to do some actions, such as an antivirus scan or maintenance.

The antivirus is able just to detect malware with a known signature, with a certain probability, or based on a heuristic. ϕ_j is the probability that the antivirus can detect (and then remove) the malware and k_j is the rate at which the scan is performed. In Tassier k, β are constants and ϕ and γ are not defined.

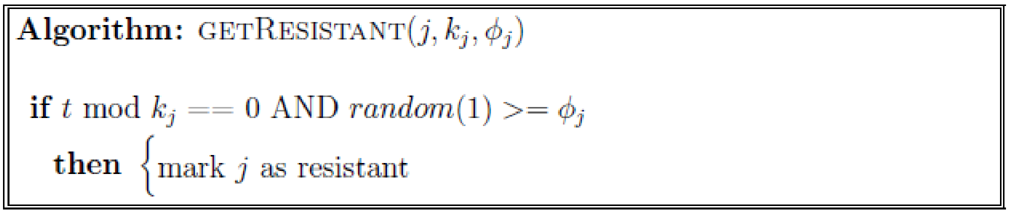
In [62,63] the spreading algorithm is:

At time t:



```
Algorithm: SPREAD($j, N_j, \beta_j, \gamma$)

 comment: j is a node

 comment: $N_j$ are the neighbours of $j$

 comment: $\gamma$ is the set of the whole gamma in the system

 for each $i \in N_j$
   do if node($j$) is infected
        then { if $random(d_j) <= \beta_j$
               then { if $random(1) >= \gamma_i$ AND $i$ is not resistant
                      then { mark $i$ as infected
```

Instead for the resistant:



```
Algorithm: GETRESISTANT($j, k_j, \phi_j$)

 if $t$ mod $k_j$ == 0 AND $random(1) >= \phi_j$
   then { mark $j$ as resistant
```

7.5.1 Netlogo

[62] implements SIR model which represents cyber disease spreading along an ICT network composed by a SCADA system interconnected to a corporate network by means of Netlogo. Netlogo [64] is a multi-agent programmable modelling environment. It provides an user interface with three tabs:

Interface tab - This tab is used both by the end-user and by the programmer. The programmer uses this tab to create buttons (e.g. for the setup and the start of the simulation) and the screen for the visualization. The end user indeed uses this tab just for see the simulation process.

Information tab - This tab is standard and not modifiable that is common for all the Netlogo's program. It can be used by an end user to gain some extra general information about Netlogo.

Procedure tab - In this tab the programmer write its code. It is composed by several procedures and some variable are sets by the interface tab with some sliders. These kinds of variable are global. In the procedure you can't pass a variable, so if you have to use a variable over several procedure, you have to declare them global.

Netlogo use three types of agents: turtles, links and patches. Mobile agents (turtles) move over a grid of stationary agents (patches). Link agents connect turtles to make networks, graphs, and aggregates. Netlogo allows the creation of sub-kind of turtles and links (called breed). A breed is a collection of agents with the same proprieties. [62] uses breed agents to group the same kind of devices (i.e. with same vulnerabilities/security policies) in order to easily set and/or to specify model variables.

7.6 Composite heterogeneous methods

At the state of the art heterogeneous modelling frameworks exist, which are able to represent cyber-attacks on Industrial Control system and SCADA, their exploitation by means of the related vulnerabilities and to evaluate the consequences of a cyber-attack on the SCADA system and in turn on the underlining CI.

An heterogeneous modelling framework hosts heterogeneous (stochastic versus deterministic, continuous versus discrete) methods:

- Combinatorial methods (i.e. Reliability Block Diagrams; Fault-trees (FT); Network Reliability Analyzers), assuming that events to be represented are statistically independent.

- State-space methods (i.e. Markov chains, Generalised Stochastic Petri Nets; Coloured Stochastic Petri Nets ; Stochastic Activity Networks) - relying on the specification of the whole set of possible states of the system and of the possible transitions among them.

- Local-dependencies methods (i.e. Dynamic Fault-trees (DFT); Bayesian Networks (BN); Dynamic Bayesian Networks (DBN)) ,assuming localized dependencies among events.

- Holistic-reductionist methods (i.e. Input-output Inoperability model (IIM); Entities-interactions simulations), basing interaction analyses on modelling input-output interactions among CIs and among CIs and their elements.

and heterogeneous tools to deal with different aspects such as i) cyber-net models to represent cyber attacks and their propagation and impact on IC S& SCADA; ii) CI models to represent and evaluate the consequences of cyber attacks on service delivered by CI to customers (i.e. power flow simulation model).

Particularly, cyber-net models defines the intrusion scenarios and its events and status. Power flow is the most basic model of the steady state behaviour of a power system. The integration of these two models makes it possible to quantify the impact caused by a potential cyber attack.

Models able to represent cyber-attacks on Industrial Control system and SCADA, their exploitation by means of the related vulnerabilities and to evaluate the consequences of a cyber-attack on the SCADA system till to the underlining CI customers, such composite models can do, generally include the following steps:

1) Model the access points to a SCADA system.

2) Construct a cyber-net model for intrusions and the status.

3) Simulate a cyber attack using the intrusion models to evaluate their impact based on CI simulations such as power flow simulations.

4) Improve cyber security of the SCADA system based on vulnerability assessment results with the available technologies.

In [88], the above method is used to assess the vulnerability of computer networks and the potential loss of load in a power system as a result of a cyber attack. Compromised cyber security of a SCADA system can cause serious impact to a power system if the attack is able to launch disruptive switching actions leading to a loss of load. This is particularly troublesome if the attack can penetrate the control center that is connected to substations under the SCADA system. The combination of access points from substation-level networks to other networks leads to various attack scenarios.

7.6.1 The Intelligent RAO Simulator

The Intelligent RAO simulator [65], [66] already used with success in the MICIE project, will also be used in the CockpitCI project for high-level, inter-system behaviour simulation as well as for validation test-bed implementation. Just as a reminder, the RAO simulator is essentially a hybrid of a discrete-event simulator, a production rules based expert system and a state-graph search based optimisation-decision making block. The system dynamics description in the simulator is based on an object oriented representation of the elements of the complex discrete system and on a description of the elements interaction (state changes) using production rules.

These formalisms don't impose limitations on the nature of simulated system and allow one to describe and to simulate various system behaviour logic. Indeed, the user describes itself classes of elements composing the system regardless of their nature, so he is free to introduce elements belonging to different infrastructures. The production rules, being the most flexible and close to human language way to express knowledge about system behaviour, also allow to describe various elements interactions including interactions of elements belonging to different infrastructures. The rules make easy control logic description, including human decision making in the process under consideration.

The RAO simulator contains a random events generation block, allowing simulation of random-event cyber attack scenarios. The observation part of the simulator includes a statistics collection system, an animation system allowing displaying the system state evolution on a computer screen, and a trace system, which writes detailed information in a dedicated file for further scrupulous analysis.

The weak point of the RAO simulator is the fact that at any event all the production rules are scanned to start possible actions, so the simulation takes a lot of processor time if the system description granularity is rather high. For example, simulating a communication network at the traffic level with all packages transmitted would take too much time related to inter-system behaviour simulation. From the other hand, a lot of specific tools exist for this kind of simulation.

So, the idea is to develop for CockpitCI project a composite simulation model including specific simulation tools for high granularity subsystems (communication infrastructure, software to a certain extent, etc.) while using the RAO simulator for intersystem behaviour representation on the higher level (cyber attack scenarios level). This concept is to be further developed in depth to precise the borders between different components of the whole simulation model.

The RAO simulator can be connected to the external world via messages processing block. Incoming external messages change the system state in accordance with programmed logic. This new state reflects in the model the change in the environment, starting a decision making. While making decision, the rules executed in the model can send in its turn messages of different types to other modules, in particular to the OPC clients linked to SCADA controlling real process. This functionality can be used to validate CockpitCI software tool. The model simulating various cyber attacks scenarios can be connected to SCADA of the test bed, thus allowing virtual testing and validation. The communication possibility can also be used to interconnect components of the whole simulation model described above.

7.6.2 Holistic reductionist method

The Mixed Holistic Reductionist approach (MHR in Figure 7.10) [67,93] is made by two layers. The first one (upper) is obtained considering each infrastructure as a whole and evaluating the impact of faults or services using domain simulators. We called this layer "holistic situation assessment". The second (lower)can be considered a reductionist impact assessment layer that is build of experts re-views and try to assess interdependencies and how faults and their consequences are reflected on other facilities.

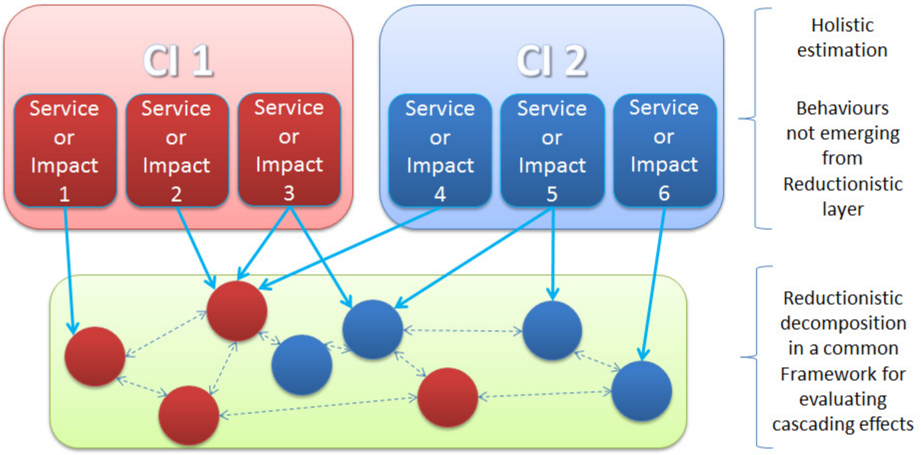


Figure 7-10 MHR model

This approach tries to guide experts in order to consider several events and not only the simple mechanical faults. In fact, in large plant, the simple physical security is not enough to protect critical infrastructures. The physical security must be assess together with cyber vulnerabilities and threats and integrating outputs of firewalls, Intrusion Detection Systems (IDS) and also information coming from national agencies and international institutions, in order to provide business continuity and the best Quality of Service (QoS) towards customers.

The definition of the term holistic is characterized by the comprehension that the parts of some-thing are intimately interconnected and explicable only by reference to the whole. In the field of Critical Infrastructure protection, evaluating faults inside the facilities where they are born using CI-related tools is a holistic point of view.

Using such point of view, a situation assessment can be realized considering several technological and organizational aspects and usually using techniques and methods specific of each facility. In the following Sections, some particular events/infrastructures will be described/analysed to better under-stand the holistic impact evaluation:

- Managing of alarms, usually collected using SCADA (Supervisor Control And Data Acquisition) software and then shown to operators in order to support decisions;
- Managing of physical security information, for detecting unauthorized accesses to specific areas, using also data mining algorithms;
- Evaluation of the Quality of Services (QoS) toward customers of the infrastructure, using simulator for analysing transients and outages after the faults [96, 97], like load-flow simulation for power grids or NS-2 for telecommunication networks;
- Detection/spreading of cyber attacks, especially worms and viruses spreading, as Red-Code [68], Stuxnet [21] or Duqu [22] worm, through mathematical representation, such as the two-factor models;
- Use of information coming from international and national agencies, like CERTs and other institutions, to integrate cyber-related data coming out from other infrastructures.

The aim of this analysis is the evaluation of the availability of both single elements of an infrastructure and the infrastructure as a whole, to provide goods and services to other elements/customers at an acceptable predefined level, in presence of faults, failures or any kind of anomaly situations.

Several simulators can be used at reductionist level. Among them NS2, the well known open source simulator, CISIA and I2SIM.

CISIA (Critical Infrastructure Simulation by Interdependent Agents) [98, 99] is an agent-based simulator for modelling critical infrastructure interdependencies. It was born with the aim to analyze failure propagation and performance degradation in systems composed of different, heterogeneous and interdependent infrastructures.

In CISIA, each facility is modelled with macro-components at a high level of abstraction. Each macro-components is defined as an agent. Each agents has the same structure based on few common quantities, representing the state or memory of the agent:

- Operative Level (OL): the ability of the agent to perform its required job. It is an internal measure of the potential production/service, if the operative level is 100% it does not mean that it is providing the maximum value but that it could, if necessary.
- Requirements (R): what the node needs to reach the maximum operative level.
- Faults (F): the level of failure that affects the agent, for each type of fault.

Agent inputs and outputs are necessary in order to perform interactions among agents.

There are three kind of inputs and, similarly, three kind of outputs:

1. Induced/propagated faults: faults propagated to the considered agent from its neighbour-hoods and from the considered agent to its neighbourhood, described in terms of type and magnitude.
2. Input/output requirements: amount of resources requested by/to other objects.
3. Input/output operative levels: the operative level of those objects whose resources are used in it, and the operative level of the object itself.

In CISIA, the agent dynamic is described as an input/output model among the previously listed quantities. This description of agent's behaviour is highly abstracted but it is enough rich in order to leave the experts to model the model dynamics in the most appropriate way.

The relations among agents are based on their interdependencies, and they are described by incidence matrices. In fact each matrix is able to spread a different type of interdependency, following Rinaldi characterization [100] among physical, geographical and cyber connection.

I2Sim is a simulation environment used to study interdependency problems in Critical Infrastructure Protection [94] and it allows to model physical and geographical interdependencies. The key element of I2Sim is the production cell, a functional unit that relates a set of resource levels as input to a particular resource level as output. Only one kind of resource can be associated to a single production cell. By considering a proper set of production cells it is possible to model a scenario consisting of different infrastructures and build loose or light coupling relations to model interdependencies.

In addition to a production cell, I2Sim also relies on other components:

- Channel: which is a means through which tokens flow from a generator cluster to a load cluster.

- Tokens: are goods and services that are provided by some entity to another entity that uses them. These tokens can be water, electricity, medical supplies, etc.

- Cluster: is a group of one or more cells. Two clusters are separated in time or space and are connected by channels. Each cluster generates and/or consumes tokens.

- Controls: are Distributor and Aggregator units. They change their state based on the events received from the decision making layer.

The generation and flow of tokens among different entities is determined by physical capability of each of the cells (e.g., power generation capacity or water supply capacity), their environmental constraint (damage of cells or delay in transportation channel) or human decision factor (e.g., redirection of water supply to a hospital rather than to a swimming pool).

Figure 7.11 shows an I2Sim electrical scenario composed by SCADA basic controlling elements and electrical substations. Two electrical substations supply energy to two different residential areas and receive commands from two RTUs, which are connected to a Public Switched Telephone Network (PSTN). It is possible to model the consequence of failures occurring inside the electrical substation at specific time, which have the effect of reducing the quantity of energy supplied to the residential areas and to the communication components. In addition, it is possible to model the consequence of a failure against a RTU (e.g. due to a cyber attack) that may reduce the quality of service of the electrical substation. This approach allows modellers to explore the impact of failures on system performance [95].

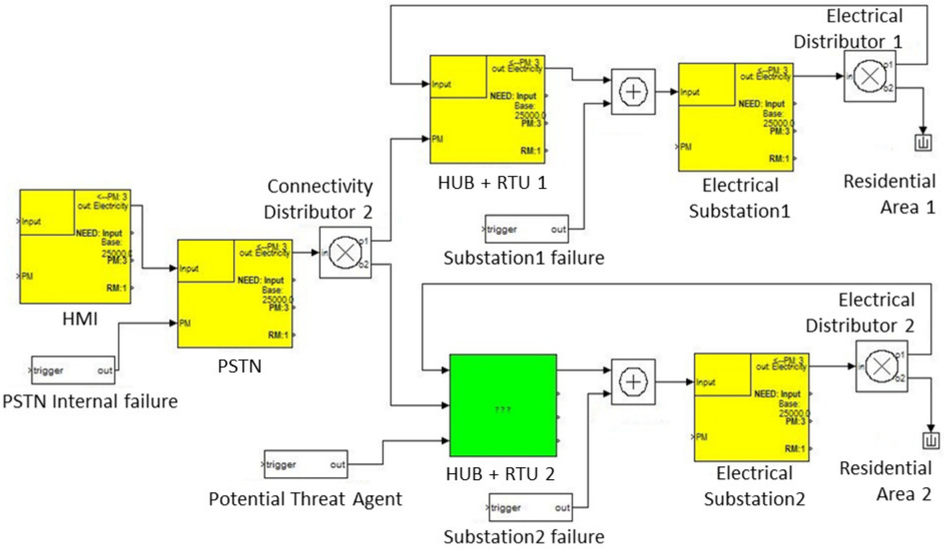


Figure 7-11 I2Sim scenario

7.6.3 Composing NS2 enterprise network model and SCADA devices

In [69] a simulation environment for analyzing and assessing the security of SCADA system and associated industrial infrastructure is described. The hierarchical structure and communication model of SCADA system is presented. A SCADA system consisting of sensors acquiring data, actuators controlling industrial devices, control devices (such as Programmable Logic Controllers (PLCs) performing logical control, central SCADA servers or Master Terminal Unit (MTU) acquiring data and sending control instructions, Human Machine Interface (HMI) displaying data for operators and providing various control input forms, database servers storing historical data, workstations engineers detecting and debugging systems, business information systems for specific industrial applications, and various types of communication devices, etc) is considered.

The simulation environment, figure 7.12, consists of several layers, mainly NS2 based simulated enterprise network, customizable OPC client/HMI, integrated industrial OPC server, extensible SCADA protocol tester, several prevalent SCADA RTUs, and the sensors and actuators flexibly deployed in specified scenario.

The components considered in the reference architecture start from the enterprise network till the industrial infrastructure. In such an architecture, network client sends data packets with certain type or format through Internet browser or special application, and the data packets are sent to OPC server or SCADA protocol tester across the simulated enterprise network. OPC server may also receive control instructions from the customized OPC client/HMI, and send the data acquired from underline industrial process to OPC client/HMI. On the other hand, SCADA protocol tester can generate the associated data units of the tested SCADA protocol through GUI or the customizable scripts, send them to lower SCADA RTUs, supervise and acquire the response data, and effectively analyze the functionality of SCADA protocol and its specification conformance and security status. The layer of SCADA RTU determines the specified RTU devices based upon the specific protocol type, receives the control instructions sent from upper layer, and controls the actuators to perform corresponding actions or controls the sensors to acquire corresponding data. The supervised industrial infrastructure usually implemented in the way of computer simulation, which supports the flexible and customizable industrial scenario simulation and specific process supervisory control and data acquisition.

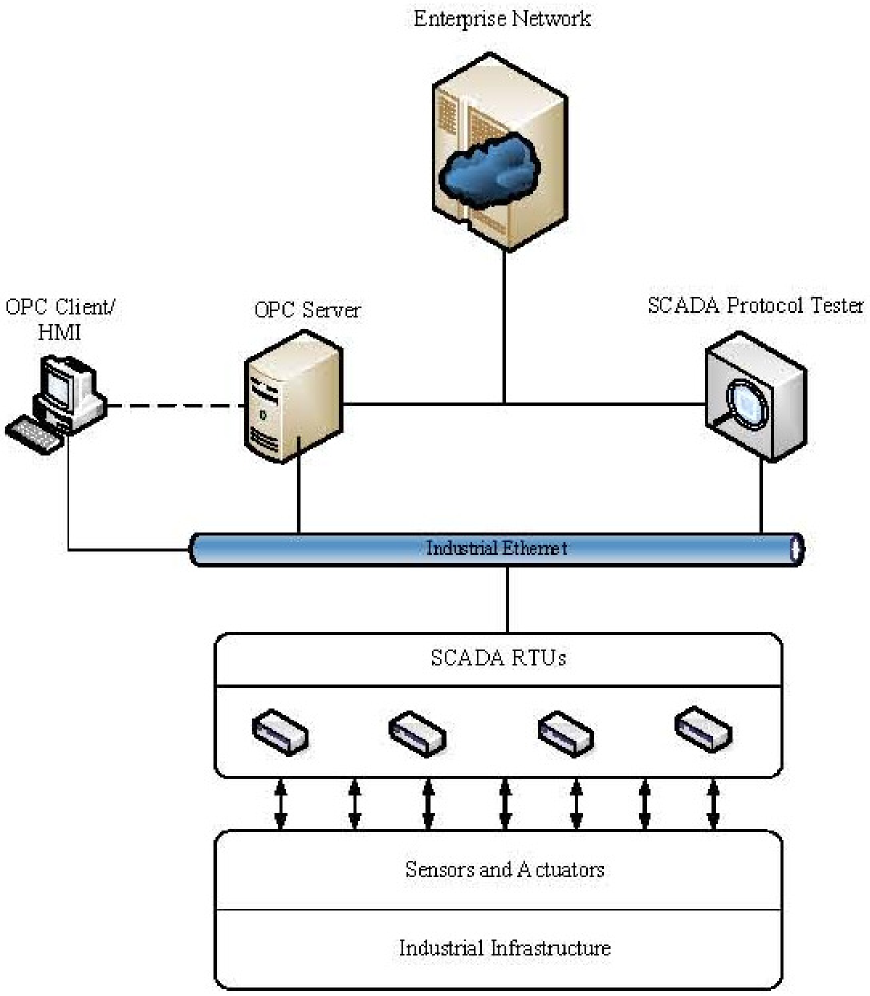


Figure 7-12 Reference Architecture of SCADA Simulation Environment [69]

Enterprise network with the appropriate scale base is simulated by NS2 emulation (NSE). Emulation refers to the ability to introduce the simulator into a live network. Special objects within the simulator are capable of introducing live traffic into the simulator and injecting traffic from the simulator into the live network. The interface between the simulator and live network is provided by a collection of objects including tap agents and network objects. Tap agents embed live network data into simulated packets and vice-versa. Network objects are installed in tap agents and provide an entry point for the sending and receipt of live data. When using the emulation mode, a special version of the system scheduler is used: the RealTime scheduler.

A TCP agent within NS2 interacts with a real-world TCP server and can receive data from the external application (figure 7.13).

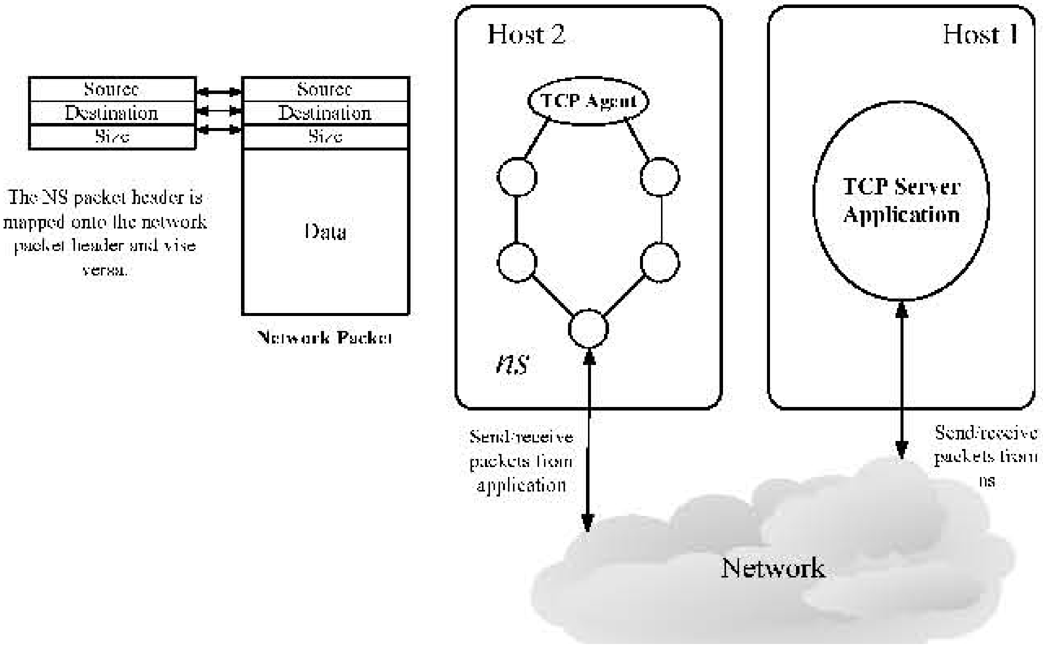

Source
Destination
Size
Source
Destination
Size
Data
Network Packet
The NS packet header is mapped onto the network packet header and vise versa.
Host 2
TCP Agent
ns
Host 1
TCP Server Application
Send-receive packets from application
Send-receive packets from ns
Network

Figure 7-13 Packets generated by TCP agent interacting with a real world TCP server [69]

OPC server acquires the industrial field data from PLC/RTUs, and send it back to OPC client application via standard OPC interfaces. OPC client application (usually refers to as HMI) displays the received industrial process data, and sends the control instructions to OPC server, which then delivers to PLC/RTUs.

As OPC server, Citect Company's CitectSCADA 6.1 has been used. The system is composed by software installed on standard computer equipment running on Microsoft Windows XP operating system, and delivers scalable and reliable supervision and control. As OPC client, they implemented an extension of open source JEasyOpc project [70], used in the experiments of security analysis and assessment of SCADA systems.

SCADA protocol tester simulates the execution of SCADA protocols based upon the protocol specifications. The protocol simulation layer provides the simulations of several SCADA protocols (including Modbus, DNP3, etc), and the functions of event scheduling and the sending and receipt of protocol packets.

Several real PLC/RTUs (GE FANUC Rx3i, VersaMax, SIEMENS S7200, S7300, ICPDAS Wincon8741 and 8ke8, etc) have been adopted, as industrial field control devices. The supervised industrial infrastructures are implemented in the way of simulation, and the environment currently could simulate a few typical industrial scenarios based upon the specified configurations, which provide the required industrial field data for the simulation system, and can respond to the control instructions. PLC/RTUs connect with sensors and actuators with digital or analog I/O, equipped with Industrial Ethernet module and multiprotocol module, and support several protocol types including Modbus-TCP, Profibus, DeviceNet, and Genius, etc.

A representative attack scenario has been developed for the experimentation. Such an attack scenario involves the following steps:

Step 1: Gain Access to Simulated SCADA System

1.1 Gain remote network access to enterprise network

1.2 Compromise the connection device between enterprise network and SCADA system

Step 2: Identify Modbus Device through Protocol Scan

2.1 Gain local SCADA access via enterprise network

2.2 Scan 502/tcp port for identifying the characteristics of Modbus device

Step 3: Compromise Master Device via Vulnerability Exploitation

3.1 Disable real slave device

3.2 Deploy rogue slave respond to Modbus requests from master

3.3 Corrupt master with invalid slave response

3.4 Load shell exploitation to master

From the above SCADA system attack experiments, the authors conclude that such a proposed environment can simulate the whole process of SCADA system attack, and provide an effective means to analyze and assess the security of SCADA system.

7.6.4 Composing Netlogo & NS2

In [62] a worst case cyber attack initiated on an ICT device and that gets out of service the redundant (primary and secondary) connections between SCADA Control Centre and its remote devices is investigated combining NETLOGO and NS2 tools.

Along the different phases of the attack the Fault Isolation and System Restoration (FISR) service, performed by SCADA, has degraded time responses which affect the quality of power to grid customers. [62] discuss a model implemented by means of NETLOGO to represent the occurrence of the cyber attack targeted at a specific system of the corporate network, the Network Management System, which spreads the infection throughout SCADA and ICT nodes up to disconnect the communication between SCADA Control Centre and its remote devices, resulting in the SCADA QoS degradation.

Model parameters include the probability of infection of a node, the virus spread rate, the intrusion detection rate of corrupted SCADA/ICT servers or remote devices and keep into account of the potentiality of the attacks, the vulnerabilities and security policies of the single SCADA and ICT elements.

The infection spreading affects FISR service and in turn the quality of power to grid customers as computed by a QoS prediction model, implemented by NS2 simulator.

The paper, with respect to the state of the art has multiform novelties: a) SCADA system is explicitly modeled with the interdependent power grid and ICT by a single model: in [3,4] a federation of multiple domain-specific simulators or tools for the simulation of interdependent infrastructures are proposed, moreover, in [6,11] cyber attacks and their propagation are not represented; b) SCADA QoS is computed against cyber attacks; limited to our knowledge there are no papers on the subject; c) modeling process is supported by two heterogeneous tools: NETLOGO focused on propagation of cyber attack effects on ICT/SCADA and NS2 which computes the impact of such effects on SCADA QoS; d) it contributes to the gap reduction between the cyber security, typically under control of ICT cyber security experts and CI operators, focused on business continuity and service availability to customers.

7.6.5 Composing Matlab & Emulab

An experimental framework to model both cyber and physical behaviour of an ICS is in [14 dipietro]. The need to propose this approach is that, although software-based simulation may be suitable to study physical systems, it has a limited applicability to cyber security because of the diverse and complex information and communications layers. The authors chose a hybrid approach that may be considered a trade-off between experimentation with real components and pure software simulation.

Modelled process

The physical layer was modelled with Matlab and corresponds to different domains: (a) a simplified version of a water purification plant with two water tanks; (b) a 160 MW oil-fired electric power plant based on the Sydsvenska Kraft AB plant in Malmo [15] which includes a boiler and turbine; (c) the previous scenario with the modelling of a condenser.

Simulation environment

The simulation testbed is based on Emulab[2] to simulate the cyber layer of an ICS such as a SCADA system. The physical layer comprises actuators, sensors and hardware devices that perform actions on the system (e.g. acquire voltages, currents etc.) whereas the cyber layer consists of all the information and communication devices and software that constitute the process network. The entire SCADA architecture can be viewed as a distributed control system consisting of the process network that usually hosts the SCADA server and the HMI and the control network that hosts PLC and other devices.

[2] Emulab (http://www.emulab.net/) is a network testbed, giving a wide range of environments in which to develop, debug, and evaluate systems. The name Emulab refers both to a facility and to a software system. The primary Emulab installation is run by the Flux Group, part of the School of Computing at the University of Utah. There are also installations of the Emulab software at more than two dozen sites around the world, ranging from testbeds with a handful of nodes up to testbeds with hundreds of nodes.

The emulation testbed allows the dynamic mapping of physical components (e.g. SCADA servers and switches) to a virtual topology. It also includes the control logic code which is implemented by PLCs in the real world.

The PLC control code can be made run sequentially with the model using a tightly-coupled code that runs in the same memory space as the model or in parallel using loosely-coupled code. The loosely-coupled code supports the execution of PLC code remotely and allows the injection of malicious code without stopping the execution of the model and the operations of more complex PLC emulators.

Simulation components

The framework has three main units:

- Simulation Core Unit: It consists of the local PLC, remoting handler and the core module. The local PLC module incorporates the PLC memory, which is used as the glue between the cyber and physical layers and the code runner module. The remoting handler module handles the communications between the local PLC modules and the local RPC system. The core module ensures the exchange of data between modules and the execution of the core timer.

- Remote PLC Unit: The main role of the remote PLC unit is to run loosely-coupled code and to provide an interface for master units to access the model.

- Master Unit: The main role of the master unit is to implement a global decision based on the sensor values received from the remote PLC units. It includes a Modbus handler module for communicating with the remote PLC units and the decision algorithm module.

Communication protocol

Communications between the simulation core and remote units are handled by .NET's binary implementation of RPC over TCP remoting feature. .NET remoting ensures minimal overhead and the use of a well-established implementation. Currently, they use Modbus over TCP for communications between remote PLCs and master units.

Implemented attacks

Although no attacks demonstration has been performed, the testbed provides an experimentation environment for understanding and measuring the consequences of cyber attacks to physical processes while using real cyber components and malware in a safe manner. Future research will focus on the analysis of the physical impact of attacks using more complex models that include descriptions of the physical components.

7.6.6 A multilevel simulation environment

In [71] is suggested a framework and a software tool that can be implemented for modelling and simulation of botnet life cycle and the process of botnet containment by defence mechanisms. Similar to the botnet structure, the structure of defence mechanisms is implemented by the subnet of particular defence components (agents). Particularly, the models of botnets and botnet defence are specified in the form of counteraction between the two classes of teams: the attack team and defence team. Each team represents a subset of computer network nodes, identified as agents, and having a common goal. The attack team includes agents belonging to the botnet and implementing actions aimed at providing the vital activities of botnets. Similarly, the defence team is made up of agents, performing the defence functions and having a common goal to oppose botnet operation.

To implement the agent-based modelling and simulation, the author intended to develop a multilevel simulation environment that differs from the well-known agent-oriented simulation tools (e.g. CORMAS, Repast, Swarm, MadKit, MASON, NetLogo, etc.) [72, 70], first of all, by the use of simulation tools that allow to adequately simulate the network protocols and security processes.

This environment is a software package that includes a discrete event simulator, implemented by low-level language, as well as a number of components that realize the components of higher levels. The architecture of the simulation environment consists of the following four main components (Figure 7.14).

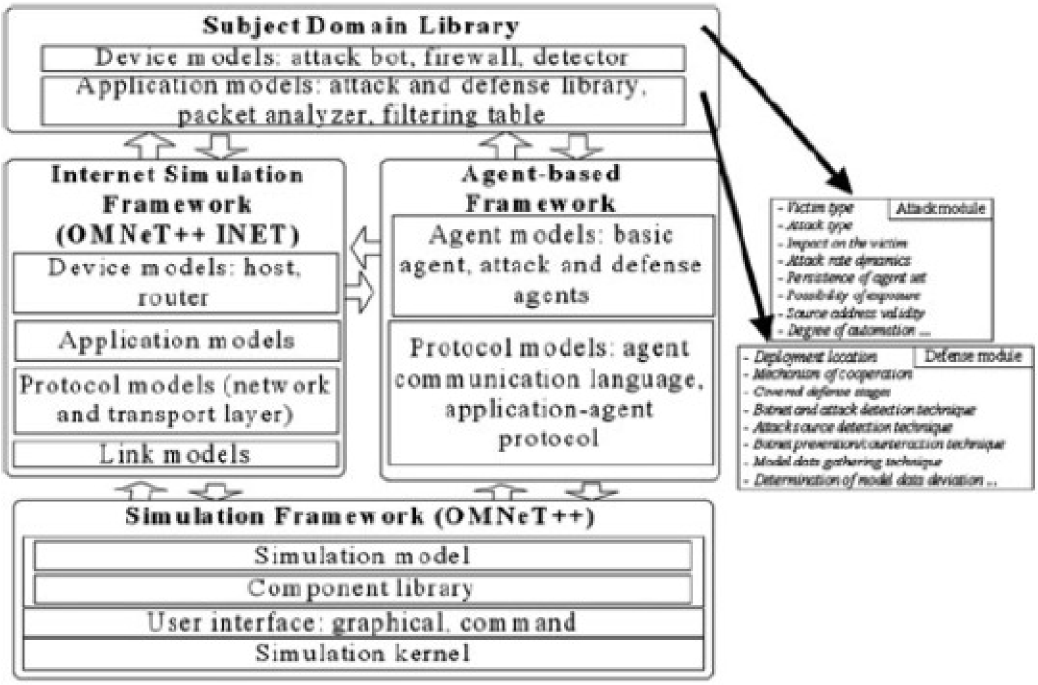

Subject Domain Library
Device models: attack bot, firewall, detector
Application models: attack and defense library, packet analyzer, filtering table
Internet Simulation Framework (OMNeT++ INET)
Device models: host, router
Application models
Protocol models (network and transport layer)
Link models
Agent-based Framework
Agent models: basic agent, attack and defense agents
Protocol models: agent communication language, application-agent protocol
Attack module
- Victim type
- Attack type
- Impact on the victim
- Attack rate dynamics
- Persistence of agent set
- Possibility of exposure
- Source address validity
- Degree of automation ...
Defense module
- Deployment location
- Mechanism of cooperation
- Covered defense stages
- Botnet and attack detection technique
- Attack source detection technique
- Botnet prevention/counteraction technique
- Model data gathering technique
- Determination of model data deviation ...
Simulation Framework (OMNeT++)
Simulation model
Component library
User interface: graphical, command
Simulation kernel

Figure 7-14 Simulation environment architecture [71]

Simulation Framework is a discrete event simulation system. It provides tools for modelling chronologically ordered sequences of discrete events. Simulation Framework implements the basic models of random event distributions and the basic models of queues with priorities and the collection of statistics. The possibilities of basic model data input/output, as well as basic features on processing the results of experiments are provided.

Internet Simulation Framework is a set of modules that allows simulation of nodes and protocols of the Internet. It contains the modules that form realistic network topologies, the models of network applications with behaviour close to the behaviour of real network applications, and the models of transport, network and link layer protocols [70].

Each protocol is implemented as an independent module. Internet Simulation Framework also contains modules for automatic construction of standard networks based on the set of defined parameters and their automatic configuration. In the current version this component uses the library ReaSE [73].

Agent-based Framework, that is a library of modules that specify intelligent agents and common scripts of their behaviour, implemented in the form of models of services and applications embedded in the models of network nodes. The component also contains the models of application layer protocols, which provide communication between agents and interaction of agents with application models. In addition, the component includes a high-level language interpreter to manage agents and a transmitting module, which converts the commands of the language into the sequence of intelligent agent actions.

Subject Domain Library is a repository that serves to simulate the processes of the subject area. It includes modules that realize different botnet and botnet defence scenarios and complement the functionality of IPnode, including filter tables, packet analyzers, models of legitimate users, etc.

In the proposed architecture, the simulation environment has been implemented by using several different components: the simulator OMNeT++ [76], the libraries INET Framework and ReaSE, and specifically developed software components [77,78].

7.6.7 Composing control station, power system and communication network models

[101] reports an heterogeneous modeling environment to demonstrate the vulnerability of the network client to a DDOS attack and the ability of filtering to mitigate an attack. The aim is to assess the vulnerabilities introduced by using public networks for communication. A network client acts s a control station, a PowerWorld server acts as a power system, and RINSE tool acts as a communication network. The attack prevented data from being transmitted across the network, causing the control display to display incorrect data.

Particularly, the main components of the environment are:

- Network client. It provides several key functions needed to implement an accurate test bed that closely mimics real world operation of the power grid. The network client (figure 7.15) provides a graphical view of power system states. The information used to drive the display is obtained via TCP/IP from a server. This mimics a SCADA HMI that is obtaining data from the field bus over a communication network.
- PowerWorld server. It is twofold:
    - simulates the power grid with a feature-rich power flow solver This allows to simulate systems with a high degree of modelling accuracy by taking advantage of the advanced modeling facilities built into the PowerWorld Simulator software.
    - provides the SCADA data that would typically be fed into a control center display (represented by the client). The server provides the simulated data to the client over a TCP/IP network using a custom networking protocol.

- RINSE (Real-time Immersive Network Simulation Environment for Network Security Exercises) tool. RINSE is a tool for realistic emulation of large networks as well as network transactions, attacks, and defenses [102]. RINSE has unique capabilities which make it suitable for cyber security and game-playing exercises including large scale real-time human/machine-in-the-loop network simulation support, multi-resolution network traffic models, and novel routing simulation techniques.

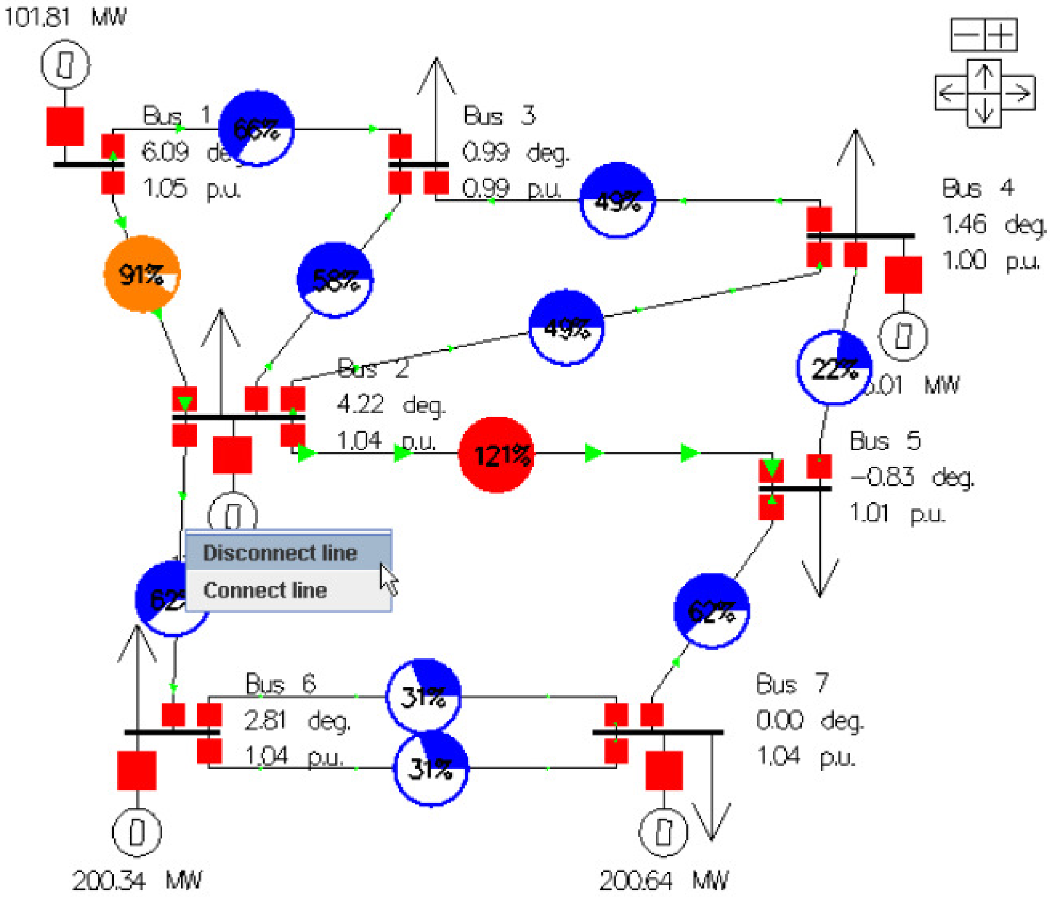


Figure 7-15 Network client screenshot - opening a line in a 7 bus case [101]

Five types of commands are currently supported by RINSE:

- Attacks: for initiating attacks (particularly DDoS attacks) in the network.
- Defenses: for applying countermeasures against attacks. These commands include filtering packets at routers which can mitigate attack effects.
- Diagnostic Tools: which simulate common networking utilities such as ping.
- Device Controls: for controlling (shutting down, rebooting) individual devices in the network

– Simulator Data: for controlling the output of the simulator.

The scheme that is used for integration of the PowerWorld simulator with RINSE is shown in Figure 7.16.

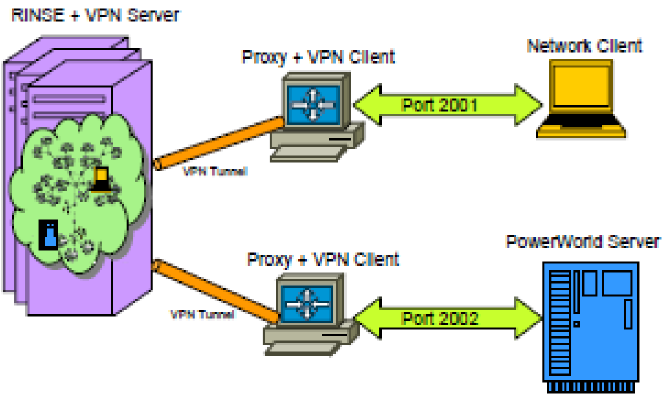


Figure 7-16 Client-Server-RINSE Integration scheme [101]

The network client sends packets to the PowerWorld server via a proxy server on a specified port. The proxy server then translates the destination of these packets to the virtual IP address of the PowerWorld server in the simulated network through a VPN tunnel. The network simulation then is performed and the same process happens in the reverse direction when the PowerWorld Server responds to the Network Client requests through the Proxy server through another VPN tunnel.

In [101], the authors performed a DoS attack considering the fact that increasing load levels cause the transmission system serving a load pocket to become overloaded. At the same time a network attack hinders the operator's ability to receive data and issue commands.

Three different scenarios have been simulated to study the effect of attack and defense. In the first scenario, the network runs under normal conditions with some transactions and background traffic present. The goal of this scenario is to study the interaction of the PowerWorld server and client under normal operating conditions of the network.

In the second scenario after a while of operating normally, a DDoS attack starts in the network. The server here is an arbitrary server in the simulated network and upon reception of the command, the attacker sends attack signals to zombie hosts in the network and they start emitting packets to the victim server at the rate of 700 Kbits/s. The attack starts after 30 seconds of simulation and lasts for 100 seconds. With this scenario we study the effect of attack on the power system that uses a public network as its communication medium.

The third scenario complements the second one by applying countermeasures in the network to mitigate the negative effects of the attack on the power grid. The intermediate router connecting the zombies network to the server network act as a filter by dropping packets coming in on all interfaces, using all protocols, from all source IP addresses and all source ports to all destination IP addresses and destination port (33333) on which the vulnerable service works. This command is issued after 60 seconds of simulation and may be described by the following pseudo-code: filter router add 0 deny all all * all * 33333.

To evaluate the attacks, an analysis of the packet drop percentage was performed in the three cases. When there is not attack occurring, the operator accessing to the network client is able to see data that is refreshed at the proper rate and have the ability to open and close lines.

If an attack is in progress the SCADA data and commands are prevented from getting to and from the network client. The DDoS attack floods the network with packets, causing the real data to be delayed or lost. This is evident in the divergence of the one-line views between the network client and the PowerWorld server. When an attack is under way, the network client continues to display old data showing that the system is operating safely even though a transmission line is overloaded. The application of a filter is one defense against a DDOS attack. Applying a filter after an attack has begun successfully mitigated the attack and allowed SCADA data to transit the network.

7.7 Security requirements modelling

Van Lamsweerde and Letier [47] and Van Lamsweerde [48] present the KAOS method for modelling security and safety requirements through the use of anti-goals (the converse of goals, such as availability and integrity). Anti-goals describe the vulnerabilities that make satisfaction of the system’s stated goals impossible.
The resulting security requirements are expressed in terms of ‘avoiding’ anti-goals in order to eliminate the vulnerabilities that would prevent the system from achieving its goals. Alexander [49] takes a similar approach of avoiding vulnerabilities, but relies on misuse cases instead of ‘anti-goals,’ as do Sindre and Opdahl [50] while McDermott [51] substitutes misuse cases for abuse cases.

8 Cyber security toolkits

Many attackers use toolkits containing different types of utilities and scripts that can be used to probe and attack systems, such as packet sniffer, port scanners, vulnerability scanners, password crackers, remote login programs and attack programs and scripts. This section reports on such toolkits with a special attention to their plugins for SCADA systems.

With the help of SHODAN's (http://www.shodanhq.com/) search engine, everyone is able to find the most vulnerable SCADA system and then create an attack for it. There are some software tools used for performing security test that can be used for malicious intent. All of this tools can be used by the attackers.

8.1 Vulnerability tools

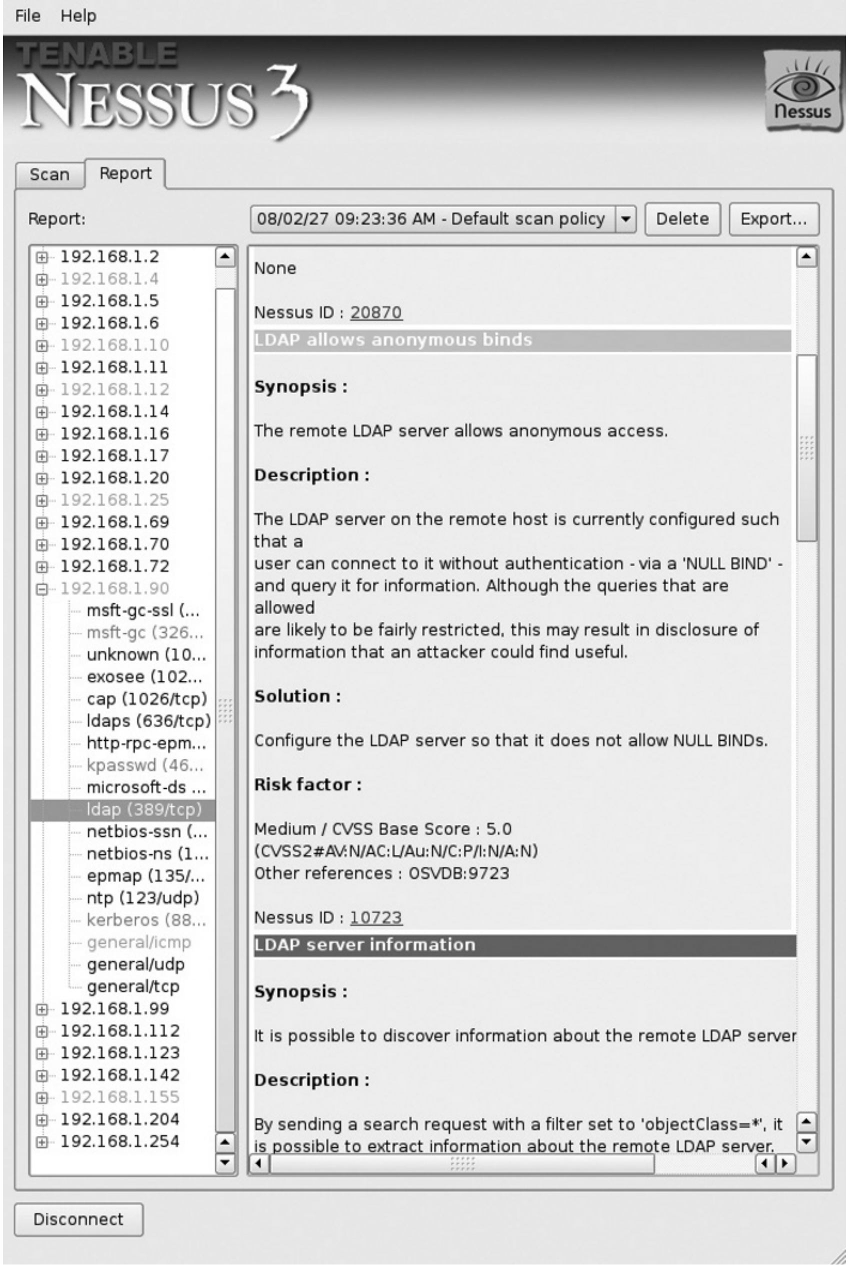
The identification of existing vulnerabilities within a network system is a big step towards staving off potential cyber-attacks that can be perpetrated against the system. Given the high complexity of today's systems, and the rapid emergence of those vulnerabilities, vulnerability tools have been perceived 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.

8.1.1 Vulnerability scanner: NESSUS

Vulnerability scanner follows a rather general pattern regardless their type. This mainly involves [81]:

1. Specifying the IP address or range of addresses to be tested
2. Detecting Live Systems by identifying which of the addresses specified by the user actually match with to running systems.
3. Identifying Live Systems by fingerprinting the types of systems that are being run at each of the previously defined addresses.
4. Enumerating Services by conducting a port scan to determine the TCP and UDP services that may be open.
5. Identifying Services at each open port. Importantly not all vulnerability tools perform this step.
6. Identifying Applications in use for each service previously identified, including their version, vendor.
7. Identifying Vulnerabilities within the applications through the use of probes/ network probes.
8. Reporting of the Vulnerabilities in a customized fashion that would enable the understanding of the management team and the system administrator.

NESSUS (http://www.tenable.com/products/nessus) can find the entire well know vulnerabilities of a system and all the misconfigured on a system and there is a library for SCADA . Tenable (the productor of NESSUS) has released 32 plugins for Nessus 3 which specifically test SCADA devices. These plugins were the result of a four month research contract between Tenable Network Security and Digital Bond (http://blog.tenablesecurity.com/2006/12/nessus_3_scada_.html).



08/02/27 09:23:36 AM - Default scan policy

192.168.1.2
192.168.1.4
192.168.1.5
192.168.1.6
192.168.1.10
192.168.1.11
192.168.1.12
192.168.1.14
192.168.1.16
192.168.1.17
192.168.1.20
192.168.1.25
192.168.1.69
192.168.1.70
192.168.1.72
192.168.1.90
msft-gc-ssl (...
msft-gc (326...
unknown (10...
exosee (102...
cap (1026/tcp)
ldaps (636/tcp)
http-rpc-epm...
kpasswd (46...
microsoft-ds ...
ldap (389/tcp)
netbios-ssn (...
netbios-ns (1...
epmap (135/...
ntp (123/udp)
kerberos (88...
general/icmp
general/udp
general/tcp
192.168.1.99
192.168.1.112
192.168.1.123
192.168.1.142
192.168.1.155
192.168.1.204
192.168.1.254

None

Nessus ID : 20870

LDAP allows anonymous binds

Synopsis :

The remote LDAP server allows anonymous access.

Description :

The LDAP server on the remote host is currently configured such that a
user can connect to it without authentication - via a 'NULL BIND' - and query it for information. Although the queries that are allowed
are likely to be fairly restricted, this may result in disclosure of information that an attacker could find useful.

Solution :

Configure the LDAP server so that it does not allow NULL BINDs.

Risk factor :

Medium / CVSS Base Score : 5.0
(CVSS2#AV:N/AC:L/Au:N/C:P/I:N/A:N)
Other references : OSVDB:9723

Nessus ID : 10723

LDAP server information

Synopsis :

It is possible to discover information about the remote LDAP server

Description :

By sending a search request with a filter set to 'objectClass=*', it is possible to extract information about the remote LDAP server.

Figure 8-1 Screenshot of Nessus tool [81]

Nessus has been developed in the early 1998 by Renaud Deraison as an open source tool, the Nessus Vulnerability Scanner has quickly established itself as one of the most used vulnerability scanner. The tool became a proprietary license in its version 3, though the engine of the tool remains free and allows for the community to release plugin updates, but charges for support and the latest vulnerability audits, including PCI, SCADA, and OS specific configurations. Nonetheless a number of independent open source project have been maintaining the **Nessus 2** engine, which contains number of plugins are still GPL. Such initiatives includes: OpenVAS and Porz-Wahn.

Practically, the Nessus vulnerability scanner [81] aims at scanning ports to detect potential vulnerabilities on the tested systems. It enables the detection of: vulnerabilities that allow a remote cracker to control or access sensitive data on a system; mis-configurations such as open mail relay and missing patches; default passwords, a few common passwords, and blank/absent passwords on some system accounts which may be easily cracked; etc. the core of its architecture revolves around three main components:
The Nessus Client and Server, the Nessus Plugins and the Nessus knowldege base.

The Nessus Client and Server adopts a client/server model, which is more convenient from a security analyst perspective. Indeed this allows the security analyst to “detach” from the vulnerability scan and use his resources for other items while Nessus continues to do what it does best [81].

The Nessus Plugins: unlike others vulnerability tools, Nessus does not rely on the release of updates from the vendor to be effective in detecting new vulnerabilities. Instead, Nessus, through its own Attack Scripting Language or NASL allows security administrators to tailor or develop their own plug in to check for vulnerabilities for the protocols and services that could be unique to their networks.

The Nessus Knowledge Base is a feature of Nessus that enables plugins to uses information already derived by primitive plugins.

Importantly Nessus uses the vulnerability metrics such as the CVSS for scoring the vulnerabilities detected during the scan.

8.1.2 Network Monitoring: Nagios

Nagios [82] is a system and network monitoring application, with the capability to provide continuous information on its posture. The application was initially developed to run under Linux system, though current extensions allow the monitoring of most Unix and window based systems. For the latter, however, this requires the installation of an agent within the Window based system that will act as a proxy between the Nagios plugin and the window services to monitor. The most common feature of Nagios includes:

- The monitoring of network services such as SMTP, POP3, HTTP, NNTP, PING, etc.
- Monitoring of host resources (processor load, disk usage, etc.)
- Existence of plugins design that can be customized to fit the need for conducting checks on specific services.
- allowing detection of and distinction between hosts that are down and those that are unreachable
- possibility to view current network status, notification and problem history, log file
- etc.

In practical term, the usage of Nagios for the monitoring begins with the definition of the host associated to the service. As depicted in the Nagios documentation (209), a host named *"remotehost*" can be created in the following way and placed in any object configuration file:

```
define host{  use generic-host ; Inherit default values from a template
              host_name remotehost ; The name we're giving to this host
              alias Some Remote Host ; A longer name associated with the host
              address 192.168.1.50 ; IP address of the host
              hostgroups allhosts ; Host groups this host is associated with
            }
```

Once this has been conducted, the next step involves the creation of the services to be monitored in that host. Specific Nagios commands can then be invocated for running existing plugins for checking services. For instance the commands check_ssh, check_http, check_smtp, check_ftp will run plugins for checking the SSH, HTTP, EMAIL and FTP server.

Upon completing the verification of a given service, Nagios issues four possible values depending on how critical a problem detected within the service is. The granularity of the alert levels are as follows:

"0" for service status being "OK"

"1" for service status being "WARNING"

"2" for service status being "CRIITICAL"

"3" for service status being "UNKNOWN"

8.2 General purpose tool: Metasploit

Metasploit (http://www.metasploit.com/about/what-is-it/) software helps security and IT professionals to identify security issues, verify vulnerability mitigations, and manage expert-driven security assessments, providing true security risk intelligence.

Capabilities include

- smart exploitation,
- password auditing,
- web application
- scanning,
- and social engineering.

Teams can collaborate in Metasploit and present their findings in consolidated reports.

Metaslpoit provides a command line interface, called msfconsole. From it , one can view the list of modules that are available. Exploit modules are used to encapsulate an exploit. Once the exploit has been selected, the next step is to determine what options it requires. Most exploits specify the target address and the target port.

Advanced option are available for some exploit modules.

Options for Intrusion Detection & Prevention System (IDPS) evasion are available too.

Once the exploit is done, there is a command to check if the target system is vulnerable to that exploit, but this is not available for all the exploit modules. When this is done, one has to define the payload, that is what the attack does.

Even for the payload module, there may have options to define. Then, everything has been done and the attack can be launched.

Metasploit framework is composed by several modules, the attacker "simply" put together such modules with some options on them and run the attack. In case of an attack on a SCADA system, some modules need to be implemented but other modules are available. A module that exploits a stack buffer overflow is available in CitectSCADA's ODBC daemon,

8.3 Sniffers: Wireshark

Sniffers, like Wireshark, are software that looks all the packages that flow through the network card. By filtering all the traffic, one can see all the relevant packages.

Doing so, and stealing the packets that have another destination, one can steal the data, password or everything written in the packages.

Another use of the sniffer is the one to mapping the whole network. It's very difficult to discover a sniffer on the net, because it does nothing (stand-alone sniffer). Even when it steals packages, it doesn't remove it, so the real destination continues to receive data (http://www.dia.unisa.it)

There are some sniffers that cause a throughput on the net, and so there are some way to detect them. The easiest ways can be easily avoided and the best solution should be "latency way" that consist to flood the network with traffic and, if the sniffer doesn't work in kernel-mode, it is slowed down and then if in the flood one insert some PING command, the computer with the sniffer responds (if doesn't lose the package) late data (http://www.dia.unisa.it).

Wireshark is able to filter and decodes DNP3 and Modbus Transmission Control Protocol (TCP). This can bring to the attacker the ability to intercept and steal information.

8.4 Exploiting MITM attacks: Ettercap

Ettercap (http://ettercap.sourceforge.net/) is a suite for man in the middle attacks on Local Area Network (LAN). It features sniffing of live connections, content filtering on the fly and many other interesting tricks.

It supports active and passive dissection of many protocols (even ciphered ones) and includes many feature for network and host analysis. It's very easy to use Ettercap in an environment with wireless communication, the only problem is to verify the compatibility with the protocol used. The lack of authentication and encryption help the hacker to use it.

All the ICS Ethernet based are subject to Address Resolution Protocol (ARP) poisoning. ARP poisoning (or spoofing) is an hack whereby an attacker sends fake ARP messages onto a LAN.

Generally, the aim is to associate the attacker's MAC address with the IP address of another host, causing a re-routing of all the traffic.

Attackers with local access to a control system subnet can use an ARP poisoning application such as Ettercap to setup and manage a MITM attack. Ettercap supports active packet content analysis and filtering. MITM attacks can be used to inject false commands, false responses, or to create a replay attack which includes false commands and false responses.

Commands may be passed from SCADA Control Centre (SCC) to RTU unchanged, may be filtered to alter command contents, or may be discarded. Similarly, responses may be passed from RTU to SCC unchanged, may be filtered to alter response contents, or may discarded.

Additionally, a MITM node does not need to wait for an actual command to initiate an attack. False commands may be issued at anytime [52]. A second MITM attack requires physical access to the SCC or RTU. In this case, a device is physically placed adjacent to them which captures MODBUS, DNP3, or EtherNET/IP transactions at their source and alters or replaces those transactions [52].

9 ICS security test beds

The meaning of ICS security test bed in literature is multifold. Two main aspects of ICS security test bed are dealt in this document:

- A program with the mission to reduce the risk of energy distruption due to cyber attacks on ICS.

- An hybrid framework composed by modelling tools and physical devices with the aim of understanding cyber security aspects of ICS and eventually, their impact on service delivered by Critical Infrastructure under control.

In any case modelling techniques and tools are largely used and integrated with physical devices and then both are accounted in this deliverable.

The details of modelling techniques and tools underlining part of such testbed are also report in chapter 7.

9.1 National SCADA Test Bed

The USA Department of Energy (DOE) Office of Electricity Delivery and Energy Reliability (OE) created the National SCADA Test Bed (NSTB) program with the mission to reduce the risk of energy disruptions due to cyber attack on control systems [103]. Since its inception, the program has formed valuable links between the government, the energy sector, and national laboratories to conduct research and development in the area of cyber security.

A key part of this mission is the assessment of ICSs to identify vulnerabilities that could put critical infrastructure at risk from a cyber attack. Although information found in individual stakeholder ICS vulnerability assessment reports is protected from disclosure, the security of the nation’s energy infrastructure as a whole can be improved by sharing information on common security problems with those responsible for developing and operating ICSs. For this reason, vulnerability information was collected, analyzed, and organized to allow the most prevalent issues to be identified and mitigated by those responsible for individual systems without disclosing the identity of the associated ICS product. Understanding the types of vulnerabilities commonly found and how to mitigate them can serve to help protect the systems currently in development as well as those already installed in ICS applications.

NSTB combines a unique range of specialized laboratory resources (Argonne National Laboratory, Idaho National Laboratory, Oak Ridge National Laboratory, Pacific Northwest National Laboratory, Sandia National Laboratories) to create realistic testing environments for SCADA communications and control systems. For example, the national test bed enables systems to undergo a level of rigorous testing that would be impossible to perform on systems in active service. The test bed provides a safe and isolated, yet real-world environment for testing and evaluating control system vulnerabilities and mitigation strategies.

Primary goals are to:

- raise industry awareness of control system cyber security vulnerability issues and mitigation techniques.

- identify, assess, and mitigate current SCADA system vulnerabilities

- develop near-term solutions and risk mitigation strategies for existing control systems.

- conduct R&D to develop next-generation architectures for intelligent, inherently secure, and dependable control systems and infrastructures.

- support development of national standards and guidelines for more secure control systems.

NSTB program was created with a clear understanding that improving the security of SCADA and control systems is integral to protecting the energy infrastructure and the sectors it serves.

[90] presents results from 24 ICS assessments performed under the NSTB program from 2003 through 2009. NSTB assessments reported large ICS attack surfaces created by excessive open ports allowed through firewalls and unsecure and excessive services listening on them. Well-known unsecure coding practices account for most of the ICS software vulnerabilities, which result in system access vulnerability or Denial of Service (DoS). However, poor patch management provides more likely attack targets because the vulnerabilities are public and attack tools are available for them. Once ICS network access is obtained, status data and control commands can be manipulated as they are communicated by unsecured ICS protocols.

Perimeter defenses cannot mitigate threats associated with required services between security zones. Vulnerabilities in Web services, database applications, and data transfer protocols can provide attack paths through firewalls. ICS network protocol applications can also be exploited to gain access to ICS hosts. Weak authentication and integrity checks allow unauthorized control or data manipulation, once ICS network access has been obtained.

NSTB assessments indicate that the biggest security threats to ICS in the energy infrastructure can be mitigated by patch management, eliminating unnecessary and unsafe services, implementing strong authentication and integrity checks to network protocols, and securing applications that accept network traffic assets.

NSTB program is based on several projects. Within the aim of this document , the following ones are considered relevant:

- Virtual Control System Environment (VCSE)

- Consequences Modelling Tool (CMT)

- Detection and Analysis of Threats to the Energy Sector (DATES)

9.1.1 Virtual Control System Environment

Given a cyber-threat, the purpose of Virtual Control System Environment (VCSE) is to help asset owners and analysts understand what effects can be achieved on control systems if the threat were to be realized, by means of a modelling and simulation tool that can be used to analyze and assess threats and cyber vulnerabilities on control systems (CS) without risking disruptions to critical operations. The tool also provides the means for evaluating selected mitigation options.

Modelling tools such as the VCSE are needed to combat the challenging technological complexities associated with securing not only legacy systems but also for the integration of emerging control system components and system architectures. As control system architectures grow in complexity and interconnectivity with other networks, exposure to more sophisticated threats, and the trend toward incorporating conventional information technology (IT) solutions, modelling and simulation tools will be needed to assist asset owners in making better-informed decisions in the selection of security solutions for their current and next-generation systems.

VCSE permits the end-user to configure a simulation environment of control system devices and network communication protocols and enable real-time, hardware-in-the-loop connectivity for the purpose of understanding the effects of cyber-vulnerabilities on CS. The VCSE will reduce the risk of energy disruption by providing a realistic setting designed to replicate portions of a vulnerable infrastructure against which cyber attacks can be played out and effective mitigation tactics developed with no threat to the actual infrastructure.

The main objectives of VCSE are:

- Develop a Regional Power System Model with enough depth and breadth to explore several scenarios and required analytics from those scenarios.
- Scenario Simulation and Analysis that develops different power system scenarios using the previous model to analyze vulnerabilities and potential mitigations to those vulnerabilities. That includes:
    - simulated attack vectors
    - appropriate visualizations to aid in analysis and explanation to interested stakeholders
    - quantitative analysis of the cyber effects and impacts

9.1.2 Consequence Modelling Tool

The purpose is to assist in the identification of threats and threat vectors against control systems by creating methods to model the physical-impact consequences that would result from a cyber attack on a critical infrastructure protection system.

The consequence analysis method and framework developed by Sandia National Laboratories and Massachusetts Institute of Technology has provided the basis for the National SCADA Test Bed Consequences Modelling Tool (CMT). This method for modelling electric power grid consequences on a local level was developed as part of a Sandia

Laboratory Directed Research and Development (LDRD) project. The method begins with a utility-ranked list of consequence categories (environment, safety, economics, etc.) and produces a value tree that represents the consequences, to a utility, which are associated with losing physical system elements. Also, the method calculates a performance index to describe the overall consequence that a threat scenario creates. Utilizing the infrastructure impact rankings generated from the consequence models helps to identify which threats are of greatest consequence to a utility. In the area of critical infrastructure protection, consequences can be defined at local, regional, and national levels. Each level can contain consequences that affect other critical infrastructures. Physical system impacts do not affect all end-users in the same way. For example, losing power to several residences for several hours may have little impact on dollar cost, but losing the same amount of power over the same several hours at an industrial plant could lead to millions of dollars in lost production; this, in turn, could lead to adverse consequences in other critical infrastructures. Understanding only the impacts associated with a particular infrastructure does not accurately depict the true loss to the asset owner. The CMT was designed to give the utility owners a picture of the total-costs associated with an outage. Before the CMT can be used, stakeholders must define a value tree that reflects importance rankings for the individual components within their system.

Based on the consequence-ranking framework, the software provides asset owners the cost—that is, loss in several dimensions—associated with an electronic power disruption; the CMT also provides an estimation of the consequences of a system failure to the serving utility at a local level. The information provided in this analysis enables 1) utility owners/operators to get the most out of their mitigation budgets and 2) cyber security providers to prioritize their development efforts.

9.1.3 Detection and Analysis of Threats to the Energy Sector

The purpose is to develop intrusion detection systems (network, host, and device-level), event correlation framework, and a sector-wide, distributed, privacy-preserving repository of security events.

Detecting cyber attacks against digital control systems quickly and accurately is essential to energy sector security. Current intrusion detection systems (IDS) continuously scan control system communication paths and alert operators of suspicious network traffic. But existing IDS, often not tailored to the control environment, typically offer limited attack response capability and frequently produce false alarms or fail to alert. Without carefully deployed monitoring, these devices can produce an overwhelming number of alarms that become difficult to correlate. This introduces system communication latency and slows incident response time.

Detection and Analysis of Threats to the Energy Sector (DATES) project aim to develop the first integrated intrusion detection, security incident/event management (SIEM), and large-scale threat analysis capability for the energy sector. DATES provides control system operators with enhanced incident detection and alerting tools through rigorous monitoring of threats at the network, host, and device levels. Integrating SIEM capabilities, the system will use attack models and information from prior events to automatically correlate alarms, distinguishing malicious cyber incidents from minor disruptions.

Additionally, utilities can lack an anonymous method to share threat information across the sector, which limits owner/operator threat visibility to what they can record on their own systems. DATES creates an anonymous global threat database, allowing utilities to securely report their threat data. System owner/operators can then view other sector security events in real time to obtain an accurate, high-level view of their security posture. Improving and integrating these security features will create a defense system against increasingly sophisticated cyber attacks.

The main objectives are:

- Integrate of IDS, SIEM with the DCS test bed.
- Implement comprehensive test mod/sim environment at Sandia with realistic process control system traffic and power model.
- Testing and refinement of DATES attack detection techniques using SRI and Sandia test environments.
- Extend/distribute models and simulations to represent larger scale systems.

9.2 PowerCyber testbed

The PowerCyber testbed, at Iowa State University (ISU), Electrical and Computer Engineering Department, provides a realistic electric grid control infrastructure based on a combination of physical, simulated, and emulated components. The testbed integrates industry standard control software, communication protocols, and field devices combined with power system simulators to provide an accurate representation of cyber-physical grid interdependencies. The testbed provides numerous cyber-security and power system research capabilities including:

- Cyber vulnerability assessment
- Attack impact analysis
- Mitigation strategy evaluations
- Cyber-physical system studies
- Vendor product interoperability
- Education and outreach

Testbed Components:

- Industry Standard Hardware/Software
- Internet Scale Event and Attack Simulation Environment (ISEAGE)
- Real Time Digital Simulator (RTDS) and DIgSILENT PowerFactory
- Mu Security Analyzer, Mu Dynamics

9.3 DETER testbed

The cyber-DEfense Technology Experimental Research (DETER) testbed is a public facility for medium-scale repeatable experiments in computer security [91,92]. Built using Utah's Emulab software (Emulab is discussed in section 7 of this document), the DETER testbed has been configured and extended to provide effective containment of a variety of computer security experiments, including defense against attacks such as DDoS, worms, viruses, and other malware, as well as attacks on the routing infrastructure.

Once registered, a security experimenter can access DETER remotely to develop, configure, and manipulate collections of nodes and links with nearly-arbitrary network topologies. The pool of testbed nodes is generally shared among multiple simultaneous experiments, isolated from each other. The node pool currently contains roughly 400 PCs, located at USC ISI and UC Berkeley but managed as a single testbed. Supported operating systems include Linux, FreeBSD, Windows, and the Click Modular Router.

Among computer security experiment, DDoS experiments explore the dynamics and effects of DDoS attacks on complex networks. They contributed to the development of a methodological framework for analyzing the effectiveness of DDoS defense technologies. This framework was refined through experiments of increasing scale and realism, using combinations of simulation, emulation on the DETER testbed, modeling, and analysis. A notational short hand was developed for describing and comparing experiments, archiving experiment descriptions, data, and results. This archive is under expansion to cover other defensive technologies and attack scenarios, and will serve as a set of resources for other DDoS experimenters, making it relatively easy for new experimenters to reuse existing software and tools to create an experiment scenario.

DDoS experiments on DETER include:

- studies of defensive technologies, using commercial and open source software, and research prototypes;
- investigation of configuration, conduct, methodology and analysis of DDoS defense, in a rigorous setting;
- examination of two specific commercial software packages: Symantec ManHunt and Network Flight Recorder (NFR) Sentivist;
- evaluation of FloodWatch, a traffic detection and response system using statistical profiling, as a defensive technology for defining, executing, and refining the experimental process.

9.4 Power systems cyber security in Italy

ENEL is a leading multinational energy company, Italy’s largest electrical power company and among the largest utility around the world.

ENEL is running a wide range of infrastructures which allow production and transmission of electricity, both at national and international level. Such infrastructures are controlled, monitored and managed by means of “Industrial Control Systems” or, more specifically, “Supervisory Control and Data Acquisition” (SCADA).

Cyber security is crucial for effective and steady operations of power systems.

One of the first steps ENEL took toward the building of a stronger cyber security posture has been the revision of its own internal security framework to make it better fitting the SCADA and control environments most common issues.

Joint research initiatives between ENEL and Global Cyber Security Center (GCSEC) are fostering comprehensive analysis of the several ICT security aspects concerning the electrical power infrastructures with the objectives of determining the most effective solutions for the extremely specific infrastructures run by ENEL.

Enel and the Italian Police recently settled an agreement to commonly develop recovery and preventative procedures to tackle cyber threats and incidents occurring to critical infrastructures. This agreement represents a further step to build a coordinated effort increasing operations protection for one of the most strategic asset of the nations.

Developing a comprehensive security program addressing all aspects of security is an ongoing process that, in many cases, requires specialized resources to keep up with the growing challenges, threats and risks coming from both inside and outside the company borders.

In ENEL of Livorno, Italy, it has been installed a laboratory in which a model replicates a SCADA for a termo-electrical power supply. As shown in figure, the backbone of the whole infrastructure is constituted by the switch L3 (named “Centro Stella”) and by firewalls, which allow the communication between the “Centro Stella” and all the connected subnetworks. The perimeter of each subnetwork is defined by the own firewall, which, in this context , works as a router. Switch L3 is physically divided into two switches which in turn, communicate throughout a big firewall the “Fortinet”. The whole network is divided into eight big areas:

1. Power Context Simulator Area. It simulates the ENEL intranet, a network with components which are outside of the production plant, but with access rights to some specific devices of the production plant. The network is composed by a Windows Domain server and two VPN clients for remote access to SCADA.
2. Horizontal Service area. It offers support services, such as Server FTP, Samba, DNS and others.
3. Threat and Attack Simulator area. It contains systems which simulate attacks.
4. Internet area. It represents an access point towards internet.
5. SCADA System area. It simulates the ICS and SCADA of the termo-electrical power supply. It is divided into two networks: the field network and the control network. Such networks represent all the links among SCADA systems within an ENEL production plant and the interfaces with other external control systems.
6. Observer Terminal area. It is composed by nodes which monitor network traffic and store information needed to evaluate the level of security.

7. Testbed Master Administrator area. It is used to remotely coordinate all the areas on which attacks are simulated.

8. Vulnerabilities, Countermeasures and Attack Repositories area. It store all the information about vulnerabilities and the related countermeasures that could be implemented to prevent possible attacks.

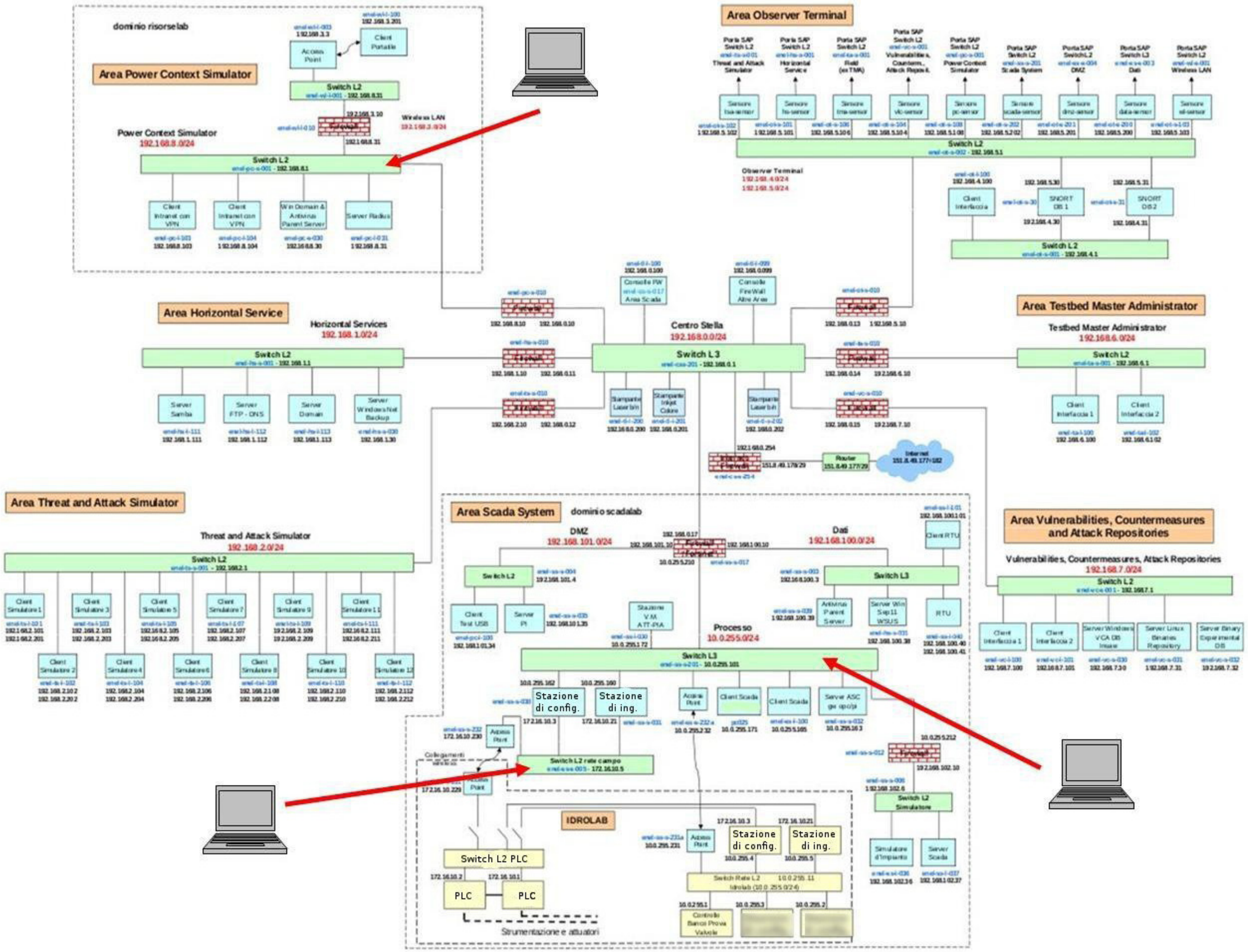


Figure 9-1 - CyberSecurity testbed laboratory of ENEL Livorno

9.5 Experimental investigation of malware attacks (MAlSim & Jade)

This section presents the results of a research on the impact of traditional ICT malware on SCADA systems [89].

In order to perform the research, the authors developed a experimental testbed that integrates malware simulation which can generally be provided by educational simulators or anti-virus testing simulators. Educational simulators are typically used to demonstrate the effects of malware infections (e.g. Hirst's Virus Simulation Suite, Virlab, etc.) whereas anti-virus testing simulators are used to simulate the behaviour of viruses in order to test anti-virus programs.

The experimental testbed recreates the architecture of a typical power plant and also incorporates software tools for conducting experiments. The main components of the power plant testbed are: the process network, the field network (hosting SCADA servers), the company network, the data exchange network, the external network (e.g. Internet) and the observer network (incorporating sensors that gather raw data about a system under attack).

The experimental testbed uses MAlSim [77, 78] which is an agent based toolkit for simulating various types of malicious software in computer networks. MAlSim is used together with JADE which provides a JAVA implementation for generic agents that is deployed over all the hosts of the experimental testbed. Each MAlSim agent class was extended with features of the malware to be simulated including the capability of: (a) making reverse engineering techniques to obtain the source code of the target malware; (b) injecting code into one of the agent classes; (c) instantiating the new malware agent in the environment.

The experiments considered well-known examples of malware i.e. Code Red, Slammer, etc. and for each of them, the source code was extracted and injected into dynamic agents to simulate the malware. Regarding the effects produced by the malware on the process network, Code Red caused the infected SCADA servers to reboot whereas Slammer infected all the exposed personal computers in few minutes.

Other experiments were implemented with malware exploiting Modbus protocol vulnerabilities, potentially causing serious damage to an industrial control system. In the one experiment, a Modbus DoS attack is programmed to discover Modbus slaves that are connected to the same network as the infected machine. Having identified the slaves, the malware sends them Modbus fabricated messages to cause DoS in order to saturate the bandwidth as fast as possible. It is important to remark that without an appropriate infection trigger, the DoS malware is effective only when the attacker is able to launch the malicious code on a computer connected to the field network or process network of a SCADA system.

Another experiment implements a Modbus worm to seize control of the slaves in the process network by exploiting the lack of authentication and integrity mechanisms in the Modbus protocol. The consequence is that when a master sends a message to a slave, the slave executes the command without checking the identity of the sender and the integrity of the message contents. Thus, any attacker with access to the network segment hosting the slaves could send forged Modbus TCP messages that force the slaves to execute unauthorized operations, potentially compromising the system.

In both of the two presented cases, in addition to attacks performed by the internal network, possible scenarios of infections involving an external attacker may be performed through email-based attacks, phishing attacks or a Modbus worm attacks.

The experimental tests have shown that malware was able to completely circumvent traditional security mechanisms by adopting ad hoc infection and attack strategies, enabling the malware to disrupt or even seize control of vital sensors and actuators. The use of encrypted channels and authentication mechanisms (e.g., those developed for DNPSec) can address man-in-the-middle attacks, but not scenarios where the master device is infected.

10 Cyber security within EU projects

10.1 AFTER

AFTER (A Framework for electrical power sysTems vulnerability identification, dEfense and Restoration) [53] 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, 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.

10.2 CRISALIS

CRISALIS (CRitical Infrastructure Security AnaLysIS) [54] 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.

10.3 PRECYSE

PRECYSE (Prevention, protection and REaction to CYber attackS to critical infrastructures) [55] 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 resilienceof 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.

10.4 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.

10.4.1 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 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 everyday 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.

10.4.2 Electricity test bed

The test bed at 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.

10.5 VIKING

VIKING (Vital Infrastructure, Networks, Information and Control Systems Management) [20] 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 skillful 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.

10.5.1 State Estimator-based Attacks and Mitigations

One of the interesting issue of Wiking project are state estimator–based attacks and mitigations.

In Figure 10.1, 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. 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.

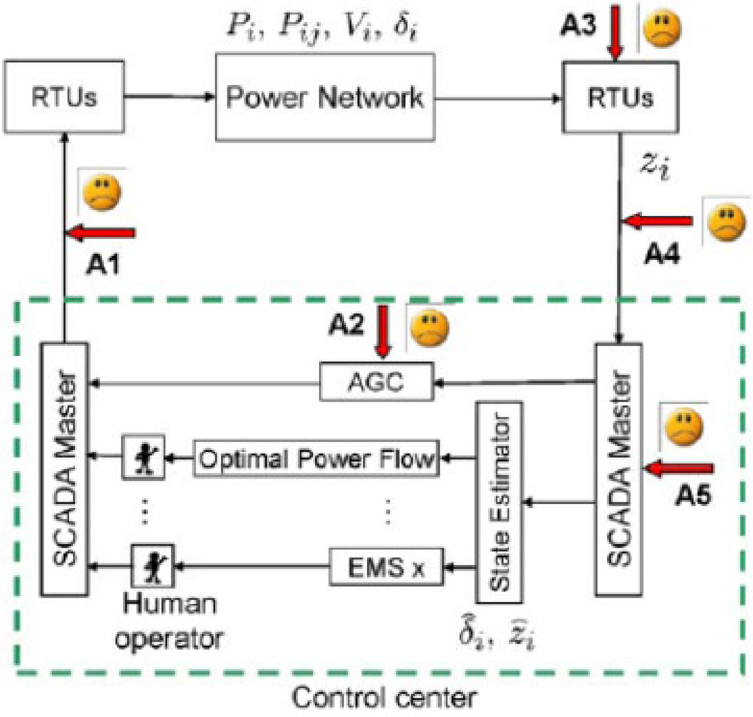


Figure 10-1 Block diagram of power network, SCADA and control center [20]

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 fiber optics, satellite, and microwave connections. Datais 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 10.1, 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.

10.5.2 CySeMoL: a Cyber security modeling 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). For instance, for the first estimate 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 figure 72% here is acquainted with a large share of uncertainty. However, in 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 get 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.

10.6 ESCoRTS

The ESCoRTS project (European network for the Security of Control and Real-Time Systems) was a EU FP7 Project started in 2008. It envolved 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 onjective 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 (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 infrastrcuture 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 analised (protocol-oriented, process-oriented, exchange network targeted), as well as several countermeasures to reduce their impact.

11 ICS cyber security standards and guidelines

NERC [79] has constituted the compliance standard CIP 1200 for a power system to meet the network security requirements [88]. 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 defense with examples [58].

As far as standards as concerned, the Common Criteria (CC) also known as ISO/IEC 15408 (ISO/ IEC, 2006a) 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 [87].

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.

11.1 NERC

NERC Standards CIP-002-3 through CIP-009-3 provide 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.

11.2 NIST

NIST. Guide to industrial control system (ics) security.

NIST. Guide to malware incident prevention and handling, 2005.

11.3 HOMELAND SECURITY

Homeland security. Recommended practice: improving industrial control systems cybersecurity with defense-in-depth strategies.

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

11.4 ISO

11.4.1 ISO 270xx

The family of standard 270xx allows to assess the security of the information system in the enterprise. Indeed, most organizations have a number of information security controls. Without an ISMS however, the controls tend to be somewhat disorganized and disjointed, having been implemented often as point solutions to specific situations or simply as a matter of convention. Security controls in operation typically address certain aspects of IT or data security, specifically, leaving non-IT information assets less well protected on the whole. In the following, some of the most important standard are presented:

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 defines an overarching security framework consisting of 133 specific controls organized around 39 control objectives. This balanced framework serves as the basis for both measuring an organization's effectiveness in addressing risk and structuring an organization's overall security program.

ISO 27032 address 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 provides detailed guidance on the security aspects of the management, operation and use of information system networks, and their inter-connections. Those individuals within an organization that are responsible for information security in general, and network security in particular, should be able to adapt the material in this standard to meet their specific requirements.

11.4.2 ISO 18043

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.

11.4.3 ISO 15408

ISO 15408 is a framework in which computer system users can specify their security functional and assurance requirements, vendors 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. 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.

11.4.4 ISO 15443

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. This will facilitate selection of the appropriate assurance method or and possible combination of assurance methods for a given IT security product, system, or service and its specific environment.

11.4.5 ISO 19791

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.

ISO 19791 is limited to the security evaluation of operational systems and does not consider other forms of system assessment. It does not define techniques for the identification, assessment and acceptance of operational risk.

11.5 ISA-99.00.02

The ISA-99 standard is designed to be general in nature and can thus be applied to any of the critical infrastructure sectors. Although it may be necessary for an organization to add specific requirements to make the standard specific to their sector or organization, the requirements in the standard can provide increased security by themselves. Specific standards for SCADA security are provided by ISA-99.00.02, "Security for industrial automation and control systems", that describes basic guidebook that an implementer of the SP99 standard can use to assemble a security program, without prescribing the details for every industry.

12 Conclusions

The document deals with an overview of modelling techniques and tools to represent ICS and SCDA system and their behaviour under cyber attacks.

Cyber security methodologies, models and tools are fundamentally based on identification of attacker profiles, attack objectives, attack steps characterization, malware spreading throughout ICS network and consequences of successful attacks on CI customers. In this view and having in mind the main objective of CockpitCi project, different cyber security methodologies, models and tools, used as a single package to address specific aspects of the attack scenario, and/or integrated together to afford the whole attack scenario are discussed.

At the state of the art, no single modelling technique has the modelling power and the analytical tractability to adequately deal with the modelling and early prediction of QoS of SCADA system facing adverse events, such as cyber attacks, and accounting cyber interdependency along CI ICT backbone. As a consequence, for analyzing ICS under cyber attacks and the related consequences on CI (i.e. Power grid) services to customers, we distinguish four kinds of models each one requiring specialized methods and tools which, in turn, could rely on specialized or not (general) modelling formalisms:

- Attacks/attacker/vulnerability models (attack/vulnerability trees, Petri nets, Game theory);
- ICS & enterprise network models (network simulators/emulators);
- CI models (i.e. electrical models by power flow simulators);
- Composite models to represent more than one aspect of the attack scenario (at least two different kinds of the previous models) till the whole attack scenario (i.e. attacks model plus ICS & enterprise network model plus CI model), which may require more than one (Hybrid versus homogeneous) method and tool.

Also, several tools which cover partially or as whole the above methods and models, have been overviewed. Many of them rely on stochastic approach such as Petri nets, Game theory, Markov chains, Bayesian networks, Monte Carlo methods; other ones rely on different approaches such as Agent based simulation, discrete event simulation, etc. Different comparison paradigms could be used to compare the modelling techniques and tools resulting from this overview.

The aim of this document is solely to offer to CockpitCI modellers a discussion about the current set of formalisms, modelling techniques and tools, used to afford the challenge of predicting the impact of successful attacks on the Industrial Control Systems (ICS), of which SCADA systems is a subset, and in turn on the Quality of Service delivered by the target CI, which is either functionally or cyber interdependent with other CIs. The final aim is to gain an enrichment of basic tools already used with success in MICIE to afford the CockpitCI vision.

Of course, modelling formalisms could be ranked according to different criteria i.e. their modelling power against analytical tractability or by their ability to represent any part of the scenario to be represented within CockpitCI project (attacker profiles, attack objectives , ICS&ICT vulnerabilities, attack steps characterization , malware spreading throughout ICS network and consequences on quality of service delivered to CI customers). A complete ranking of modelling formalisms, techniques and tools, that will be exercised on CockpitCI reference scenario, will be provided at a later stage of the project within the appropriate WP2000 deliverables.

13 References

[1] NIST. Guide to industrial control system (ics) security.

[2] NIST. Guide to malware incident prevention and handling, 2005.

[3] D. H. Ryu et al. , Reducing security vulnerabilities for critical infrastructure, Journal of Loss Prevention in the Process Industries, 2009.

[4] Homeland security. Recommended practice: improving industrial control systems cybersecurity with defense-in-depth strategies.

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

[7] 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.

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

[10] 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.

[11] Pollet, J. (August 8, 2002). SCADA Security Stratege, PlantData Technologies, http://www.plantdata.com/ SCADA/Security/Strategy.pdf.

[13] Christofer Minich and Howard Ragunton. Surviving a cyber attack on your SCADA system.

[14] Rose Tsang. Cyberthreats, vulnerability and attack on SCADA network.

[15] Raoul Chiesa. Profiling hackers: real data, real experiences, wrong myths and the hacker profiling project (hpp), 2009.

[16] M. Grimes, “SCADA Exposed,” ToorCon 7, 2005.

[17] Triangle MicroWorks, Inc, DNP3 Overview, Raleigh, North Carolina, 2002, http://www.trianglemicroworks.com/documents/DNP3_Overview.pdf

[18] Byres Research, British Columbia Institute of Technology, OPC Security Whitepaper #2 OPC Exposed May 1, 2007.

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

[20] 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.

[21] L. O’Murchu N. Falliere. W32.Stuxnet dossier, Symantec White Paper, February 2011.

[22] W32.Duqu. The precursor to the next Stuxnet, Symantec White Paper, November 2011.

[23] McAfee Foundation professional service and McAfee Labs. Global energy cyberattack: "night dragon", 2011

[24] http://www.opcfoundation.org/.

[26] ICS & Smart Grid Cyber Security, Workshop on the Framework for 'ICS and Smart Grids Testing & Certification' Paris, April 20th 2012

[27]A. Avizienis, J. C. Laprie, Brian Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, 1:11–33, January-March 2004.

[28] F. Cohen, Managing Network Security - Attack and Defense Strategies. Network Security Magazine, July 1999.

[29]Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog. On Stochastic Modeling for Integrated Security and Dependability Evaluation. The Journal of Networks (ISSN 1796-2056), Vol. 1, No. 5, September/October 2006.

[31] W. Reisig, Petri Nets: an Introduction, Springer Verlag, 1985.

[32] J.L. Peterson, Petri Net Theory and the Modeling of Systems, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[33] Mattew H. Henry, Ryan M. Layer, Kevin Z. Snow, and David R. Zaret. Evaluating the risk of cyber attacks on SCADA systems via petri net analysis with application to hazardous liquid loading operations. IEEE, 2009.

[34] Penet tool. http://powercyber.ece.iastate.edu/penetintro.html

[35] G. Ciardo, J. Muppala, and K. Trivedi, User Manual for SPNP: Stochastic Petri Net Package.

[36] http://www.micie.eu

[40] Igor Kotenko, Alexey Konovalov and Andrey Shorov,Agent-based simulation of cooperative defence against botnets, CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE (2011)

[41] Macal C, North M. Tutorial on Agent-based Modeling and Simulation. The 2005 Winter Simulation Conference, Orlando, FL, USA, 2005.

[42] Macal C, North M. Tutorial on agent-based modeling and simulation part 2: how to model with agents. The 2006 Winter Simulation Conference. Monterey, California, USA, 2006.

[43] Mahadevan P, Krioukov D, Fomenkov M, Huffaker B, Dimitropoulos X, Claffy K, Vahdat A. Lessons from Three Views of the Internet Topology. Technical Report, Cooperative Association for Internet Data Analysis (CAIDA),2005.

[44] OMNeT++ Community Site. Available from: http://www.omnetpp.org/. 2011.

[45] Kotenko I, Ulanov A. Agent Teams in Cyberspace: Security Guards in the Global Internet. The International Conference on CYBERWORLDS (CW'06). Lausanne, Switzerland, 2006.

[46] Kotenko I, Konovalov A, Shorov A. Simulation of Botnets: Agent-based approach, Intelligent Distributed Computing IV. Studies in Computational Intelligence, Vol. 315. Springer-Verlag, Berlin, 2010.

[47]Van Lamsweerde A., Brohez S., De Landtsheer R., Janssens D. (2003) 'From System Goals to Intruder Anti-Goals: Attack Generation and Resolution for Security Requirements Engineering', Proceedings of the Workshop on Requirements for High Assurance Systems Workshop, Monterey Bay, CA, September.

[48] Van Lamsweerde A. (2009) Requirements Engineering: From System Goals to UML Models to Software Specifications, John Wiley & Sons Ltd, West Sussex, England.

[49]Alexander I. (2002) 'Modelling the Interplay of Conflicting Goals With Use and Misuse Cases', Proceedings of Workshop on Goal-Oriented Business Process Modeling, London, pp.1-7, CEUR-WS.org

[50] Sindre G. and Opdahl A..L. (2000), 'Eliciting Security Requirements by Misuse Cases', Proceedings of the 37th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS), Sydney, Australia, November

[51] McDermott J.J. (2001), 'Abuse Case Based Assurance Arguments', Proceedings of the 17th Annual Computer Security Applications Conference, New Orleans, IEEE computer society

[52] Wei Gao, Thomas Morris, Bradley Reaves, and Richey Drew. On SCADA control system command and response injection and intrusion detection, 2010.

[53] http://www.after-project.eu/.

[54] http://www.ctit.utwente.nl/research/projects/international/fp7-streps/crisalis.doc/

[55] http://cordis.europa.eu/projects/rcn/61016_en.html.

[57] http://www. scada testbed/sandia/ciattacks.pdf

[58] A. Torkilseng and G. Ericsson, "Some guidelines for developing a framework for managing cybersecurity for an electric power utility," ELECTRA Report , Oct. 2006.

[59] Chee-Wooi Ten and Chen-Ching Liu. Vulnerability assessment of cyber security for SCADA systems. IEEE, 2008.

[61] T. Tassier. SIR Model of Epidemics, Anual report, 2005.

[62] E. Ciancamerla, M. Minichino, S. Palmieri - On prediction of QoS of SCADA accounting cyber attacks - Probabilistic Safety Assessment and Management Conference (PSAM11) and the Annual European Safety and Reliability Conference (ESREL 2012) - Helsinki, Finland - 25-29 June 2012

[63] E. Ciancamerla, A. Di Pietro, C. Foglietta, M. Minichino, S. Palmieri, S. Panzieri- From Holistic Assessment to Impact Evaluation- in CRITIS, 7th International Conference in Critical Information Infrastructures Security, 2012

[64] http://ccl.northwestern.edu/netlogo/.

[65] MICIE Project, Deliverable D2.2.1: Interdependency modelling framework, interdependency indicators and models, pp 68-70.

[66] Introduction to Intelligent Simulation: The RAO language. A. Artiba, V.V. Emelyanov, S.I. Iassinovski. Kluwer Academic Publishers, 1998.

[67] S. De Porcellinis, S. Panzieri, R. Setola, "Modelling critical infrastructure via a mixed holistic reductionistic approach," Int. Journal of Critical Infrastructures, Inderscience eds., vol. 5, n. 1/2, pag. 86-99, Inderscience,

[68] C. C. Zou, W. Gong, and D. Towsley. 2002. Code red worm propagation modeling and analysis. In Proceedings of the 9th ACM conference on Computer and communications security (CCS '02), Vijay Atluri (Ed.). ACM, New York, NY, USA, 138-147.

[69] Wang Chunlei, Fang Lan, Dai Yiqi, A Simulation Environment for SCADA Security Analysis and Assessment,International Conference on Measuring Technology and Mechatronics Automation,2010

[70] JEasyOpc project. http://www.sourceforge.net/projects/jeasyopc

[71] Kotenko I, Konovalov A, Shorov A. Simulation of Botnets: Agent-based approach, Intelligent Distributed Computing IV. Studies in Computational Intelligence, Vol. 315. Springer-Verlag, Berlin, 2010.

[76] Gamer T, Scharf M. Realistic Simulation Environments for IP-based Networks. The First International Workshop onOMNeT++, Marseille, France, 2008.

[77] R. Leszczyna, I. N Fovino, M. Masera, Simulating malware with MAlSim ,J Comput Virol (2010) 6:65–75

[78]R. Leszczyna, I. Nai Fovino, M. Masera, MAlSim: Mobile agent malware simulator, in: Proceedings of the First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, 2008.

[79] “Cyber security standards,” NERC. [online]. Available: http://www.nerc.com/~filez/standards/Cyber-Security-P ermanent.html, 2006.

[80] CockpitCI Project, Deliverable D5.1: CockpitCI System requirements

[81] Rogers R., Carey M., Criscuolo P. and Petruzzi M. (2008) Nessus network auditing 2nd edition, Syngress Publishing, Inc. Elsevier, Burlington.

[82] Nagios documentation (2009) http://nagios.sourceforge.net/docs/nagios-3.pdf

[83] C. Xiaolin, T. Xiaobin, Z. Yong, and X. Hongsheng. A markov game theory-based risk assessment model for network information systems. International conference on computer science and software engineering, 2008.

[84] P. Liu, W. Zang, and M. Yu. Incentive-based modeling and inference of attacker intent, objectives, and strategies. ACM Transactions on Information and System Security (TISSEC), 8(I):78-l 18, 2005.

[85] Q. Wu, S. Shiva, S. Roy, C. Ellis, and V. Datla. On Modeling and Simulation of Game Theory-based Defense Mechanisms against DoS and DDoS Attacks.SpringSim 2010.

[86] J. Jormakka and J. V. E. Molsa. Modelling information warfare as a game. Journal of Information Warfare; Vol. 4(2), 2005.

[87] 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

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

[89] I. Nai Fovino, A. Carcano, M. Masera, A. Trombetta. An experimental investigation of malware attacks on SCADA systems. International Journal of Critical Infrastructure Protection, 2(4): 2009.

[90] NSTB Assessments Summary Report: Common Industrial Control System Cyber Security Weaknesses, May 2010, INL/EXT-10-18381

[91] Benzel, Terry, Bob Braden, Dongho Kim, Clifford Neuman, Anthony Joseph and Keith Sklower Ron Ostrenga and Stephen Schwab, Experience with DETER: A Testbed for Security Research. Second IEEE Conference on testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom2006), March 2006, Barcelona.

[92] Faber, Ted, John Wroclawski, and Kevin Lahey, A DETER Federation Architecture In Proceedings of the DETER Community Workshop on Cyber-Security and Test, August 2007, Boston.

[93]E. Ciancamerla, A. Di Pietro, C. Foglietta, M. Minichino, S. Palmieri, S. Panzieri- From Holistic Assessment to Impact Evaluation- in CRITIS, 7th International Conference in Critical Information Infrastructures Security, 2012.

[94] H. Rahman, M. Armstrong, D. Mao and J. Marti, "I2Sim: A matrix-partition based framework for critical in-frastructure interdependencies simulation," in Electric Power Conference (EPEC), Vancouver, 2008.

[95] A. R. Haz, K. Beznosov, J. R. Marti, Identication of Sources of Failures and their Propagation in Critical Infrastructures from 12 Years of Public Failure Reports, International Journal of Critical Infrastructures (IJCI), Vol. 5, No. 3, pp. 220-244, 2009.

[96] E. Ciancamerla, M. Minichino, V. Rosato, G. Vicoli - SCADA systems within CI interdependency analysis: cyberattacks, resilience and quality of service - Workshop on Experimental Platforms for Interoperable Pub-lic Safety Communications - Joint Research Centre (JRC) – 10, 11 October 2011- Ispra – Italy

[97] E. Ciancamerla, C. Foglietta, D. Lefevre, M. Minichino, L. Lev and Y. Shneck - Discrete event simulation of QoS of a SCADA system interconnecting a Power grid and a Telco network - 1st IFIP TC11 International Conference on Critical Information Infrastructure Protection 2010 World Computer Congress 2010 proceedings - Springer - Brisbane 2010 – ISSN 1868-4238

[98] S. De Porcellinis, S. Panzieri, R. Setola, "Modelling critical infrastructure via a mixed holistic reductionistic approach," Int. Journal of Critical Infrastructures, Inderscience eds., vol. 5, n. 1/2, pag. 86-99, Inderscience, 2009.

[99] S. De Porcellinis, S. Panzieri, R. Setola, G. Ulivi, "Simulation of Heterogeneous and Interdependent Critical Infrastructures," Int. Journal of Critical Infrastructures, vol. 4, n. 1/2, pag. 110-128, Inderscience Ent.. Ltd., UK, 2008

[100] S. M. Rinaldi. Modeling and Simulating Critical Infrastructures and Their Interdependencies. In Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 2 - Volume 2 (HICSS '04), Vol. 2. IEEE Computer Society, Washington, DC, USA, 20054.1-, 2004.

[101] Grier, T. Overbye and D. Nicol, SCADA cyber security testbed development, Proceedings of the Thirty- Eighth North American Power Symposium, pp. 483–488, 2006.

[102] M. Liljenstam, J. Liu, D. Nicol, Y. Yuan, G. Yan, and C. Grier, “Rinse: the real-time immersive network simulation environment for network security exercises,” in In Workshop on Principles of Advanced and Distributed Simulation, 2005.

[103]http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/FY09_Work_Plan_External.pdf

[105]C.A. Petri, UCommunication with Automatan, Tech. Rep. RADC-TR-65-377, Rome Air Dev; Center, New York, NY, 1966.

[106]NIST. Guidelines for smart grid cyber security: Vol. 1, smart grid cyber security strategy, architecture, and high level requirements.

[107]U.S. department of homeland security. Introduction to control systems cyber security.

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

[109]Dimitrakos T, Raptis D, Ritchie B and Stølen K. (2002) 'Model based security risk analysis for webapplications', Proceedings of Euroweb 2002, British Computer Society.

[110]ISO/IEC (2006a), 'Common criteria for information technology, Part 1-3, Version 3.1, ISO/IEC15048,Geneva, Switzerland.

[111]ISO/IEC (2006b) Information technology -- Security techniques -- Security assessment of operational systems', ISO/IEC Technical Report 19791, Geneva, Switzerland.

[112]Jajodia S., Noel S., and O'Berry B. (2005) 'Topological analysis of network attack vulnerability'. Massive Computing 5(3), pp. 247-266, Springer, doi: 10.1007/0-387-24230-9_9

[113]Manadhata P.K & Wing J. M. (2010) 'An Attack Surface Metric', IEEE Transactions on Software Engineering, doi: 10.1109/TSE.2010.60

[114]Mell P., Grance T. (2011) The NIST Definition of Cloud Computing, Special Publication 800-145 (Draft), National Institute of Standards and Technology, Information Technology Laboratory, Gaithersburg, MD

[115]Ouedraogo, M., Savola, R., Mouratidis, H., Preston D, Kadraoui, D., Dubois, E. (2011) 'Taxonomy of quality metrics for assessing assurance of security correctness', Software - Quality Journal, Springer DOI: 10.1007/s11219-011-9169-0

[116] 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 Software 85(1), pp. 193-208.

[117]Slade R.(2006), Dictionary of Information Security, Syngress Publishing Inc, Canada

[118]Swanson M. (2001) Security Metrics guide for Information Technology System, National Institute of Standards and Technology, Special publication #800-26, Gaithersburg, MD

[119] Swanson M., Nadya B., Sabato J., Hash J. and Graffo L. (2003) Security Metrics Guide for Information Technology Systems, National Institute of Standards and technology Special publication 800-55, Gaithersburg, MD.

[120]T. Sommestad, M. Ekstedt, L. Nordström, A case study applying the Cyber Security Modeling Language. In Proceeding of CIGRE (International Council on Large Electric Systems), 2010

[124] Qiang Ye, Mike H. MacGregor, Combining Petri Nets and ns-2: A Hybrid Method for Analysis and Simulation

[130]Karin Sallhammar, Bjarne E. Helvik and Svein J. Knapskog. On Stochastic Modeling for Integrated Security and Dependability Evaluation. The Journal of Networks (ISSN 1796-2056), Vol. 1, No. 5, September/October 2006.

[150] 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

[151] NISCC, NISCC good practice guide on firewall deployment for SCADA and process control networks.: National Infrastructure Security Cordinaton Centre (NISCC), February 2005.

[152] IEEE Standard for Local and Metropolitan Area Networks: Port based Network Access Control, IEEE 802.1X-2004, December 2004.

[153] 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

[154] OSSEC HIDS. Available at: http://www.ossec.net

[155] 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.

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

[157] Staniford-Chen S, Tung B, Schnackenberg, D. The Common Intrusion Detection Framework (CIDF). Proceedings of the 1998 Information Survivability Workshop, ISW'98, 1998.

[158] Tung B, et al. The Common Intrusion Detection Framework Specification, 2001.

[159] Kahn C, Porras P, Staniford-Chen S, Tung B. A Common Intrusion Detection Framework, 1998.

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

[161] 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.

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

[163] 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.

[164] 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.

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

[166] Soule A, Salamatian K, Taft N. Combining filtering and statistical methods for anomaly detection. Proceedings of the Internet Measurement Conference 2005, IMC'05, 2005.

[167] Lakhina C, Crovella M, Diot C. Diagnosing network-wide traffic anomalies. Proceedings of ACM SIGCOMM'04, 2004.

[168] 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.

[169] 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.

[170] Silveira F, Diot C. Urca: Pulling out anomalies by their root causes. Proceedings of the 29th IEEE Conference on Computer Communications, INFOCOM'10, 2010.

[171] 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.

[172] 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.

[173] 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.

[174] 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.

[175] 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.

[176] 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.

[177] 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

[178] 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

[179] Snort IDS. Available at: http://www.snort.org

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

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

[182] Industrial Defender, Inc. company site: http://www.industrialdefender.com

[183] Nessus vulnerability scanner. Available at: http://www.tenable.com/products/nessus

[184] Metasploit penetration testing. Available at: http://www.metasploit.com/

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

[186] Immunity software, Inc., Canvas penetration testing framework. Available at: http://www.immunitysec.com

[187] SCADAHacker, SCADA Modules for Metasploit. Available at: http://scadahacker.com/resources/msf-scada.html

[188] Secure Crossing, Inc., Zenwall product family. Available at: http://www.securecrossing.com/our-products/

[189] Peterson, G., “Product Review Part II – Industrial Defender ASM Online Demo”, Digitalbond. Available at: https://www.digitalbond.com/2012/02/16/product-review-part-ii-industrial-defender-asm-online-demo

[190] Wireshark project. Available at: http://www.wireshark.org/

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

[192] Spitzner, Lance. Honeypots: Tracking Hackers. Addison-Wesley Professional, 2002.

[193] Provos, Niels. "A Virtual Honeypot Framework." In Proceedings of the 13th USENIX Security Symposium. 2004. 1-14.

[194] 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.

[195] Honeyd – Virtual Honeypot, http://www.honeyd.org/

[196] 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

[197] Shelia client honeypot, available at: http://www.cs.vu.nl/~herbertb/misc/shelia/

[198] 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.

[199] Radek Hes, Peter Komisarczuk, Ramon Steenson, and Christian Seifert. The Capture-HPC client architecture. Technical report, Victoria University of Wellington, 2009.

[200] Lance Spitzner et al: The Honeynet Project: Research Alliance, http://www.honeynet.org, Honeynet.