Erich Rome
Robin Bloomfield (Eds.)

LNCS 6027

Critical Information Infrastructures Security

4th International Workshop, CRITIS 2009
Bonn, Germany, September/October 2009
Revised Papers

Springer

Lecture Notes in Computer Science 6027

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Erich Rome Robin Bloomfield (Eds.)

Critical Information Infrastructures Security

4th International Workshop, CRITIS 2009
Bonn, Germany, September 30 – October 2, 2009
Revised Papers

Springer

Volume Editors

Erich Rome
Fraunhofer IAIS
53754 Sankt Augustin, Germany
E-mail: erich.rome@iais.fraunhofer.de

Robin Bloomfield
City University, London, Centre for Software Reliability
Northampton Square, London, EC1V 0HB, UK
E-mail: reb@csr.city.ac.uk

Library of Congress Control Number: 2010930333

CR Subject Classification (1998): B.4.5, C.2, K.6.5, D.4.6, J.1

LNCS Sublibrary: SL 4 – Security and Cryptology

ISSN 0302-9743
ISBN-10 3-642-14378-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-14378-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

This volume contains the proceedings of the 4th International Workshop on Critical Information Infrastructures Security (CRITIS 2009). The workshop was held from September 30 to October 2 in the Günnewig Hotel Bristol in Bonn, Germany. The workshop was organized by the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), Sankt Augustin, Germany.

CRITIS 2009 continued the series of successful CRITIS Workshops. Companies, research institutions, and governmental organizations from all main areas of critical infrastructures took an active part in supporting CRITIS and we found CRITIS 2009 both exciting and informative. The selected papers addressed a range of key issues and demonstrated the ubiquity and global importance of information infrastructures. Each paper had at least three independent technical reviews and we accepted 13 full papers out of 34 submissions.

We were very fortunate in having a range of invited speakers that covered policy, research and industry perspectives. James Smith from Los Alamos National Laboratory addressed the challenges and achievements in their work on "Large-Scale Modeling and Simulation of Critical Infrastructure." Orestis Terzidis, Vice President SAP AG, talked on the "The Internet for Energy: Perspectives and Challenges." Continuing with the energy theme, Alla Heidenreich, from SIEMENS AG, Corporate Research and Technologies (Germany) provided her insights on the "Secure ICT Infrastructure for the Future Power Grid."

Critical infrastructure protection is an area where an effective private–public partnership is required. A government perspective was provided by Michael Pilgermann, German Ministry of the Interior, who talked on German strategy regarding CIIP. Another important perspective on the partnership was provided by Paul Nicholas, Director of Global Security Strategy, Trustworthy Computing, Microsoft Corporation, who discussed "Reliance, Risk and Resiliency and the Role of Public Private Partnerships" and the changes required in how governments and industry partners understand risk, improve deterrence, respond to incidents and promulgate trust across the ICT ecosystems.

A successful workshop relies on many people. We would like to express our gratitude to all the members of the IPC who provided us with over three reviews per paper and were crucial to establishing the technical quality of the event. In addition the Local Arrangements Chairs, Rüdiger Klein and Uwe Beyer, of Fraunhofer IAIS, Germany were instrumental in providing the critical infrastructure for the workshop itself. We appreciated the overall guidance from our General Chairs Stefan Wrobel, Fraunhofer IAIS and University Bonn, Germany and Costas Lambrinoudakis, University of the Aegean, Greece and the input from the Sponsorship Chair, Bernhard Hämmerli, Hochschule Luzern, Switzerland.

In times of a global economic crisis, the Sponsorship Chair had a difficult task to perform. We are happy that we were able to win the companies IABG and Elsevier, as well as the DIESIS project, as official sponsors of CRITIS 2009 and that we received ideal support from IFIP, JRC, and the German Federal Office for Information Security, too. This support is gratefully acknowledged, as is the effort of the Sponsorship Chair Bernhard Hämmerli.

CRITIS 2009 was a truly international event, attracting 67 authors and participants from all over the world, though – not surprisingly – Europeans had the majority. Interestingly, C(I)IP is not exclusively a concern of developed countries. A number of papers from developing countries were presented at CRITIS 2009. For instance, a case study from South Africa demonstrated an effective method for assessing C(I)IP possibilities particularly for developing countries with less developed infrastructures and smaller budgets for their protection.

We very much valued the variety of talks and discussions at CRITIS 2009 and hope that these proceedings provide a lasting insight into the contribution of the workshop to understanding critical information infrastructures.

February 2010

Erich Rome
Robin Bloomfield

CRITIS 2009

Fourth International Workshop on
Critical Information Infrastructures Security

Günnewig Hotel Bristol
Bonn, Germany
September 30 – October 2, 2009

Organized by
Fraunhofer Institute for Intelligent Analysis and Information Sytems (IAIS)

Program Co-chairs

Erich Rome	Fraunhofer IAIS, Germany
Robin Bloomfield	City University London and Adelard LLP, UK

General Chairs

Stefan Wrobel	Fraunhofer IAIS and University of Bonn, Germany
Costas Lambrinoudakis	University of the Aegean, Greece

Sponsorship Chair

Bernhard M. Hämmerli	Acris GmbH & University of Applied Sciences Lucerne, Switzerland

Local Organization Chairs

Uwe Beyer	Fraunhofer IAIS, Germany
Rüdiger Klein	Fraunhofer IAIS, Germany

International Program Committee

Fabrizio Baiardi	Università di Pisa, Italy
Sandro Bologna	ENEA, Italy
Stefan Brem	Swiss Federal Department of Defense, Civil Protection and Sport, Switzerland
Emiliano Casalicchio	Università di Tor Vergata Rome, Italy
Roberto Clemente	Telecom Italia, Italy

Geert Deconinck	Katholieke Universiteit Leuven, Belgium
Giovanna Dondossola	Cesi Ricerca, Italy
Myriam Dunn	ETH Center for Security Studies Zurich, Switzerland
Erol Gelenbe	Imperial College London, UK
Stefan Geretshuber	IABG, Germany
Adrian Gheorghe	Old Dominion University, USA
Stefanos Gritzalis	University of the Aegean, Greece
Nouredine Hadjsaid	L.E.G. – Grenoble Institute of Technology, France
Bernhard M. Hämmerli	Acris GmbH & University of Applied Sciences Lucerne, Switzerland
Rüdiger Klein	Fraunhofer IAIS, Germany
Pierre-Dominique Lansard	France Telecom, France
Paul Lewis	Network Security Innovation Platform, UK
Javier Lopez	University of Malaga, Spain
Eric Luiijf	TNO DefenceSecurity and Safety, The Netherlands
Marcelo Masera	Joint Research Centre European Commission, Institute for the Protection and Security of the Citizen, Italy
Simin Nadjm-Tehrani	Linköping University, Sweden
Eiji Okamoto	University of Tsukuba, Japan
Ciaran Osborn	Centre for the Protection of National Infrastructure, UK
Dirk Reinermann	BSI, Germany
Roberto Setola	Università CAMPUS Bio-Medico, Italy
Sujeet Shenoi	University of Tulsa, USA
Neeraj Suri	TU Darmstadt, Germany
Salvatore Tucci	Università di Tor Vergata Rome, Italy
Paulo Verissimo	Universidade de Lisboa, Portugal
Stephen D. Wolthusen	Royal Holloway, University of London, UK, UK & Gjøvik University College, Norway
Stefan Wrobel	University of Bonn and Fraunhofer IAIS, Germany
Jianying Zhou	Institute for Infocom Research, Singapore

Local Organization Committee

Birgit Dorn, Yvonne Grabowski, Torsten Heinrich, Achim Kapusta, Christine Malich, Ulrich Nütten, Daniela Plath	Fraunhofer IAIS, Germany

Steering Committee

Chairs

Bernhard M. Hämmerli	Acris GmbH & University of Applied Sciences Lucerne, Switzerland
Javier Lopez	University of Malaga, Spain
Stephen D. Wolthusen	Royal Holloway University, UK & Gjøvik University College Norway

Members

Sandro Bologna	ENEA CR-Casaccia, Rome, Italy
Sokratis Katsikas	University of Piraeus, Greece
Erich Rome	Fraunhofer IAIS, Germany
Roberto Setola	Università Campus BioMedico Roma, Italy

Table of Contents

On Modelling of Inter-dependent Network Infrastructures by Extended Leontief Models

Gregorio D'Agostino, Roberto Cannata, and Vittorio Rosato

ENEA - Ente per le Nuove tecnologie Energia e Ambiente
Centro Ricerche "Casaccia",
Via Anguillarese 301, 00123 S.Maria di Galeria (Roma), Italy
http://www.progettoreti.enea.it

Abstract. We report on recent developments on stochastic chain modelling of inter-dependent network infrastructures. The problem is approached in the spirit of the Leontief tradition within the framework of the Inoperability Input/Output Modelling [6] where, instead of dealing with the sector products, inoperabilities are introduced to describe the status of the global system.

A Markov Chain evolution law replaces Leontief equilibrium condition upon external changes, thus allowing to follow the transition from an equilibrium configuration to an other and possibly mimic cascade effects triggered by unwilled disturbances. Moreover, as a variation to the "System of Systems" approach, each network is not treated like an holomorphic entity, but its inner structure is inspected. Multiple implementations of the same scenarios at different level of granularity have been compared providing evidence for intrinsic inconsistency of models disregarding the geographic distribution of networks.

Keywords: Stochastic Chains, I/O models, IIM, Inter-Dependence, Leontief Model.

1 Introduction

Complex Systems (CS), in our current understanding, are dynamical systems where active elements (nodes) interact to each other via physical (or simply logical) links. The interplay of the morphological (topological) properties and the nature of interactions produces a variety of behaviors which have prompted the CS metaphor to the attention of a large scientific community [1], [2]. An impressive number of dynamical systems have been analyzed by using the CS metaphor, allowing to unveil striking analogies between physical systems differing for their nature, extent and function. The network of worldwide flight interconnections and that of bacterial metabolism have been analyzed under common standpoints and revealed impressive morphological analogies. These studies have pointed out to the understanding, among others, of the selective pressure driving these systems to spontaneously assume a specific topology able to ensure a robustness against faults and to provide large efficiency in their functioning.

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 1–13, 2010.

Analysis of CS has slowly spread from the field of "natural" systems to that of "technological" ones. Large technological objects, in many cases, fall into the category of Critical Infrastructures (CI's). Electrical power transmission and distribution networks, telecommunication (data, voice) networks, roads, oil and gas pipelines etc. are objects whose fate might have a significant impact on daily life and, often, on humans well-being. In this respect there is a large deal of efforts in applying ideas and methods of CS to them, particularly to study their vulnerability and their response to fault. The main aim is to increase their resilience and to reduce the effects that a fault, regardless of its accidental or intentional origin, might produce.

A further level of complexity which cannot be neglected when dealing with CI's is produced by the extent of *inter-dependence* among them. Although a true hierarchy in the CI's set cannot be properly established, it is evident that a fault (producing a loss or a degradation of services) on one CI might have more or less strong impact on other CI's, and producing, for instance, service degradation up to cascade effects simultaneously affecting many CI's. A large and prolonged electrical blackout might have relevant repercussion on the functioning of telecommunications, transports etc.; degradation in the services delivered by these latter undermines the possibility of restoring the faulted network thus producing a negative feedback on the whole system of systems. Electrical networks are controlled via telecommunication systems: when these are strongly degraded following a severe electrical blackout, they cannot be used to restore the electrical network thus producing a sort of "deadlock" inhibiting (or making difficult) its recovery. The study of the effects induced by system's inter-dependence is, thus a key issue in the area of technological systems; a major expectation is the possibility of assessing analysis and control strategies through cross contamination of the different CI's improvements.

In the middle of last century, the Nobel graduate W. Leontief, introduced his celebrated matrices (or tables) to quantify economic sectors production dependencies [3]. Since that time a lot of efforts (see for instance the recent work [7]) have been devoted to provide simple models to predict macroscopic evolutions of complex systems. Along with this line of thought, this paper aims at providing a simple model to describe network inter-dependence to predict the basic macroscopic evolution of events, including cascades.

2 Extended Leontief Models

From the theoretical standpoint, the so-called Leontief approach represents a class of models which have been intensively used for the study of interdependency effects in economical systems. That approach is an Input-Output model where macrosectors are connected through a matrix (the Leontief matrix) which includes the topological structure of the interdependency and its extents.

$$Q_i(t) = \sum_{j=1}^{N} L_{ij} Q_j(t) + D_i(t). \tag{1}$$

In the original works the Q_i variables were the "sector products" and the D_i were external disturbances to the system [5]. The basic idea has been extendend to the degradation of network performances by means of the core notion of *inoperability*, a generic term indicating a variable measuring the expected percentage of a system's incomplete function. I/O models based on inoperabilities are commonly referred to as "Interdependence Input/Output Models (IIM) [6]. Hereafter the Q_i will represent the i-th "inoperability" and all the systems we will be considering will be described by a set of N inoperabilities $Q_1 \dots Q_N$. No extra quantity will define the status of the system. It is worth stressing that even such a simple approach may provide non trivial insights on the basic mechanisms driving system response against unwilled events; whereas more complex and accurate abstraction models may possibly hidden the origin of phenomena.

From the mathematical point of view, the inoperability of a system (or a component) is a numerical variable, spanning the [0,1] range, reflecting the capability to provide the function it is devoted to. The null value corresponds to complete operability, while the unitary value represents complete inoperability.

Moving from IIM original approach, we have explored possible extentions by introducing three basic new features: Time evolution, Stochasticity and Inner structure of networks. To the best of our knowledge, none of such variations is absolutely new, however they have never been integrated as it is proposed in the present work. The inner structure of the systems has also been inspected by [8] within the Petri Net framework and by [9] by means of the holistic approach.

As mentioned, we have replaced Leontief I/O relations with a Markov Chain model for the inoperabilities. Therefore our abstraction of the whole system is represented by an N-ple of variables (Q_i) assuming values in an N dimensional unitary hypercube. In the vectorial representations the general evolution equation may be written as:

$$Q(t') = \psi_t(Q(t)); \tag{2}$$

where ψ does not represent a mere function, but a stochastic rule to determine inoperabilities at subsequent time. The first order expansion of the "function" ψ leads to an equation close similar to Leontief model:

$$Q(t') = H(t)Q(t) + \Xi(t) + R(o^2(Q)). \tag{3}$$

In the present paper higher order corrections will be neglected, but they are subject of ongoing work. Under the linearity assumption, the evolution equation will acquire the simple form:

$$Q(t') = H(t)Q(t) + \Xi(t). \tag{4}$$

This equation exhibits differences and similarity with eq. 1 (original Leontief model) that are worth noting. At any instant t, the matrix $H(t)$ and the term $\Xi(t)$ are stochastic variables that may assume different values, while Leontief matrix is a fixed set of numbers. The inoperabilities at the right hand side refer to a subsequent time and not to the same time as in the native Leontief Model. Equilibrium condition for the former equations 4 are in strict correspondence with Leontief quantities:

$$Q_{eq} = E[H]Q_{eq} + E[\Xi]; \tag{5}$$

where the symbol $E[.]$ indicates the expectation value and Q_{eq} represents the set of inoperabilities at equilibrium. The Leontief matrix L and the disturbance are the expectation values of the corresponding stochastic quantities in our model.

$$\begin{cases} L = E[H], \\ D = E[\Xi]. \end{cases} \tag{6}$$

Despite the simplicity of the approach, the model may simulate progressive depletion of each component operability and the consequent degradation of the QoS's (Quality of Service) of the networks. Since system evolution must take place inside the unitary N-dimensional hypercube, the value of the matrix H and the constant term Ξ are subject to constraints. We have overcome this limitations by introducing a cut-off, thus partly loosing linearity.

External disturbances (the constant term in the equation) of different shape and duration are applied and released at will. Such disturbances, as well as interdependence coefficients, have been given a stochastic nature. The dominant behavior of the external disturbances is assumed to be deterministic, however a noise is superimposed and the coefficients of the Leontief matrix are allowed to fluctuate around their average values.

In the "System of Systems" approach, the overall system is described as a net where each node corresponds to a single infrastructure. On a N-nodes system, if one defines the *inoperability* of each infrastructure as $Q_i(t)$ $(i = 1, \ldots, N)$ and with G_{ij} the transition matrix, the general dynamical response of the system can be expressed as:

$$\frac{dQ_i(t)}{dt} = \sum_{j=1}^{N} G_{ij}Q_j(t) + \phi_i(t). \tag{7}$$

Previous equation represents a, continuous time, deterministic evolution equation. By integrating the former equation in a finite time interval $(t' - t)$ one obtains a linear integral equation close to classical I/O models:

$$Q(t') = T(e^{\int_t^{t'} G(s)ds}Q(t)) + \int_t^{t'} T(e^{\int_t^s G(s')ds'})\phi(s)ds; \tag{8}$$

where $T()$ symbol indicates time ordered integration. Under the stationarity hypothesis, (i.e. when the generator G is time-independent) one gets a simplified linear equation:

$$Q(t') = e^{(t'-t)G}Q(t) + \int_t^{t'} e^{(t-s)G}\phi(s)ds. \tag{9}$$

Therefore, apart from time discretization and the non trivial possible dependence of matrix L on time, the systemic and the Leontief approach coincide (by posing $e^{(t'-t)G} = L$).

An other important limit in the "system of systems" native approach is that networks (or the services they provide) are regarded as a global entity without entering their inner morphology. In this paper we have tried to deal with the internal structure of the networks expanding the former single node into a subnet. "Network to network" interactions are then replaced by "component to component" interactions: like looking at the previous picture by means of a magnification lens (see fig. 1).

We have foreseen two different types of inner structure. One based on the geographical disposition of devices and an other based on the internal organization, at service level. Nevertheless outcoming results do apply to any internal organization regardless of its origin.

Concerning the geographic structure, we have introduced a "locality" constraint: the components belonging to different networks are allowed to interact each other (i.e. to depend on) when they are located at the same site only; while components belonging to the same network interact according to their usual functional relations. We have called such approximation "Local

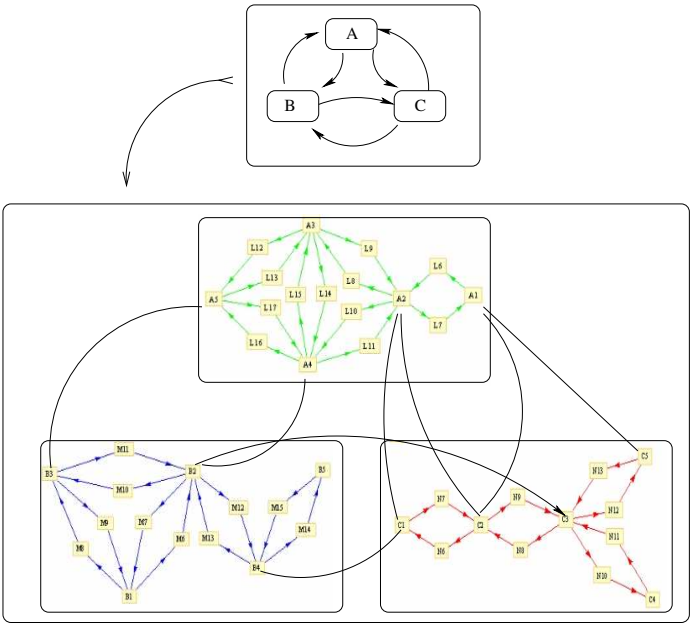


Fig. 1. Pictorial representation of Leontief model in the "System of Systems" perspective (left side), compared with our extended model. "Energy Power", "Telecommunication" and "Highways" are labelled with capital letters A, B and C, respectively. A1, A2, ... , B1, B2, ..., C1, C2, ... label components localized in single site (such as power plants), whereas L1, L2, ..., M1, M2, ..., N1, N2, ... represent linear extended (components such as cables); according to Local Dependence Approximation, dependences between components of different nets are allowed for site components only.

Dependence Approximation" (LDA). On the other hand, when dealing with internal organizations of different origins, the geographic constraint above does not make sense and all to all dependences are allowed.

Hereafter, to enhance legibility, component inoperabilities will be indicated by the lower case symbols q_i while the capital letters Q_i will be reserved for the network global inoperability. Therefore eq. 4 in terms of components will read:

$$q(t') = hq(t) + \xi(t); \tag{10}$$

whereas at the lower level of granularity the equation reads:

$$Q(t') = HQ(t) + \Xi(t); \tag{11}$$

We have assumed that coefficients of the interdependence matrix H are "a priori" known. In principle, those coefficients should be fitted on real system data measured during unwilled events, such as outages, storms or wide range ICT attacks. The privacy policy of the companies very often makes such a program tremendously difficult and in fact we were not able to collect required data.

3 Typical Scenarios

This section is devoted to report the three basic different scenarios that may be described by means of our stochastic chain simulation model. We will make distintion between disruptive and non disruptive disturbation depending on whether they are able to lead one component of the system (or more of them) to a completely inoperable state or not. We will also distinguish between stable and unstable systems.

To be definite we will focus on a simple system consisting of three interdependent service networks let say 'Energy Power", "Telecommunication" and "Highways".

The Leontief approach to that system may be graphically represented by a graph consisting of three nodes (each representing a service network) and six arcs linking all pairs of them. Fig. 1 (left side only) pictorially shows that approach. Non trivial coefficients of Leontief matrix are associated with the six "represented" (drawn) arcs. The same picture shows (right side) the pictorial representation of our enhanced model where each network is expanded into its components: the first 17 components refer to the "Energy Power"; components from the 18-th to 32-nd belong to the "Telecommunications" network; and components from the 33-rd to 45-th originated from the "Highways" network. LDA hypothesis is imposed to the system; therefore, interactions among components belonging to the same network reflect the network internal organization, while cross interactions (cross dependence) are only allowed when components are positioned at the same geographical site.

In all the forthcoming scenarios, an external disturbance will be applied to the system at time-step 500 (in arbitrary time step units) in order to reproduce

the sudden onset of the unwilled event (regardless of its accidental or deliberate origin), whereas such disturbance will be removed at time-step 1300, thus mimicing a suited human action or the end of a natural event.

3.1 A Non Disruptive Disturbance

Depending on the stochastic characteristics of the matrix h, the system response to an external disturbance may vary. Fig. 2 shows the whole set of 45 components behavior as a function of the discrete time for a stable system under a "non disruptive" unwilled event.

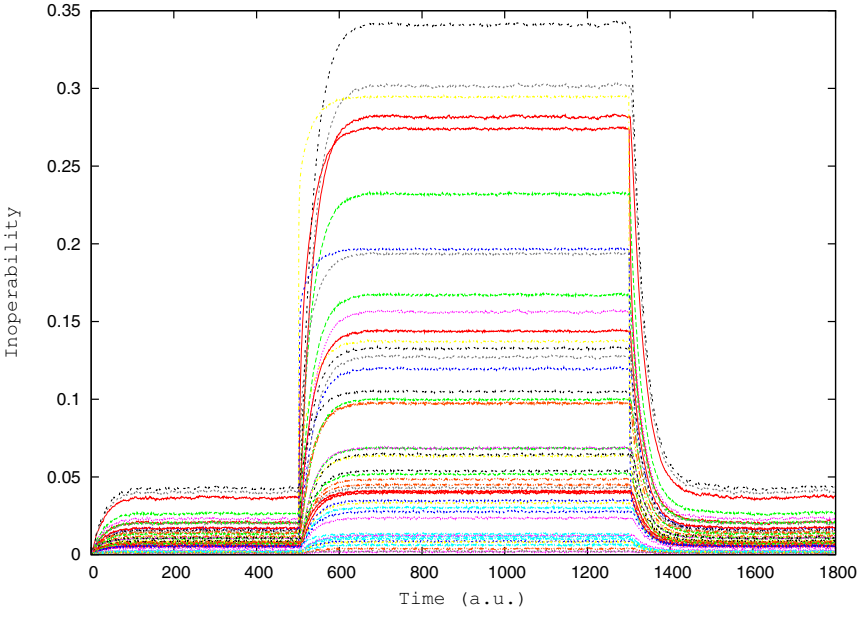


Fig. 2. The inoperability profiles of the whole set of 45 components defining our synthetic system

The system starts in a completely operable state at initial time (t=0). At that time, the noise is activated. The noise is represented by a positive definite stochastic variable (with a typical level of 1 per thousand inoperability) uniformly distributed among all components. Due to this soft perturbation, the system acquires non trivial average inoperability values. Fig. 3 shows the first 200 steps of the entire trajectory when the system reaches the equilibrium condition due to noise. Evidently this part of trajectory is an artefact reported only to provide evidence for the mere noise effects.

At time-step 500, an external deterministic constant disturbance is applied to component 33, that is the first component of the electric system. This event

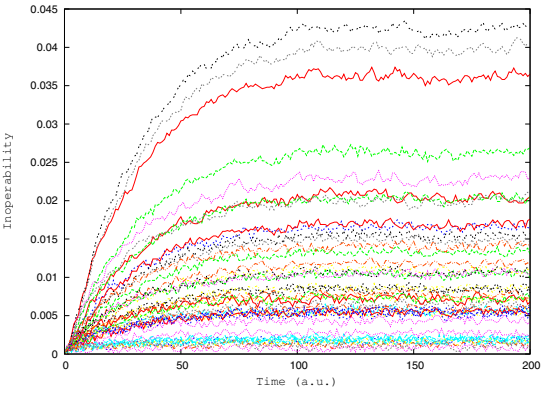


Fig. 3. The inoperabilities of the whole set of 45 components defining our synthetic system during the first stage of the simulation when noise is introduced, but no external disturbance is applied

represents some partial malfunction of the electric system at the first geographic location. Due to intrinsic limits of our modelling we are not able to further qualify the nature of the external disturbance nor of the affected component. As a consequence of the external disturbance, the system is brought to a new equilibrium condition where all components acquire non trivial inoperabilities. It is worth noting that, as for the initial noise, the system spends about 200 time-steps to achieve the new equilibrium condition. Moreover none of the components become completely inoperable (Q=1). Finally the most disturbed component is not the one suffering for external direct disturbation, thus providing evidence that the system responds globally to the local disturbation. Fig 4 (left side) shows the first 100 time-steps after the advent of the undesiderable event.

As already mentioned, we assume that the external disturbance is switched off at step 1300. This may simulate an effective human intervention or the spontaneous release of the external source. In both the cases, the system restores its original levels of inoperabilities. Since the system exhibits a non disruptive equilibrium configuration under mere noise application and a complete recovery upon external disturbance release, we may qualify it as "stable". Fig. 4 (right-side) shows first 100 time-steps of the "recovery stage".

So far we have treated the system at the highest available level of granularity that is we have followed all components during their evolution. From the macroscopic point of view, in several cases, one is only interested in the global state of health of each service network. One may define the operability of a network as the average operability of its component with suitable weights depending on the their relevance:

$$Q_s \stackrel{def}{=} \frac{\sum_{k \in C_s} q_k \cdot w_k}{\sum_{k \in C_s} w_k}; \tag{12}$$

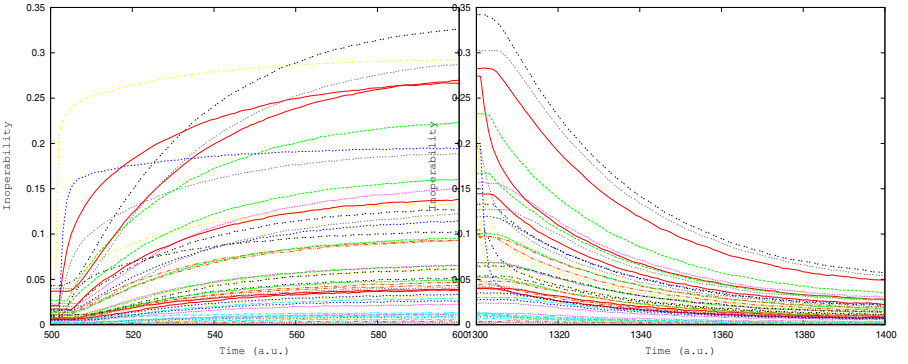


Fig. 4. The transients to the new equilibrium condition under an external disturbance of 10% inoperability of the Electric Power system. The left side represents the onset of the disturbation, whereas right side represents the restoration upon disturbance release.

where C_s represents the set of components belonging to the s-th network, q_k's are component inoperabilities and w_k's are relative weights of the different components. Such weights are assumed to be all unitary as the system we are treating is just an artifact, but in principle appropriate weights should be given. When observing the behavior of the system (under the same disturbation as in fig.2) at the low level of granularity, we are left with the typical macroscopic behavior reported in fig. 5. Evidently this picture provides a synthetic representation of the global status of the system while lacking on geographic information.

The procedure of going from the higher level of granularity to the lower is immediate and easy to perform; unfortunately such a procedure is actually "ill-defined". To provide evidence of this crucial point, we have disturbed the electric power of the same system (i.e. same matrix) at the same time and with the same level of external disturbance, but varying the component on which the external disturbance is applied. Unfortunately, we are left with a rather different pictures (fig. 5): the equilibrium values of inoperabilities in the second case, are, in fact, higher. Moreover the rates of the global inoperabilities among the different service networks at equilibrium are different. This observation provides evidence of the intrinsic limits of the model (especially at low granularity level) as the effect of a fault in a network depends on the specific component it is affected by. The phenomenon is well known among experts and, in fact, they are rather skeptical about low granularity simulations.

The scenario depicted above represents the typical non disruptive disturbance applied to a stable system. In fact, no component has ever reached the complete inoperability state and the system is capable to reach a novel equilibrium state upon the external disturbance application. This capability of the system is related to the characteristics of the probability distribution of matrix h. The relation between the average algebraic properties of the matrices and the response of the system is far from the purposes of present work, nevertheless it

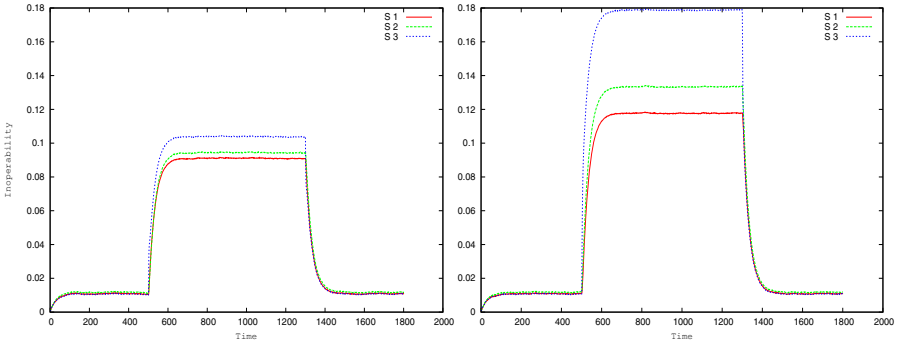


Fig. 5. The global inoperabilities of the different services in the simulated non disruptive scenario. A ten percent disturbance is applied to the Electric Power (i.e. third service). Picture on the left side refers to a case when fault takes place on the first component of the electric system, while the picture on the right side refers to a fault on the third.

represents ongoing work. We just mention here that, when the sum of the rows of the matrix h are bounded to be less than unitary, the system it represents is always stable.

3.2 A Disruptive Disturbance

When increasing the level of external disturbance, or when the system is more reactive, one or several components may experience the complete inoperability. Fig. 6 (left side) shows the inoperability profiles of all components. Two of them are brought by the external disturbance to a unitary level of inoperability, that is they are completely "out of order" until the cause of the original disturbance is removed at time 1300. It is worth stressing that the initial disturbance operates over a component that reduces its operability, but never ceases to work; on the contrary, two other components are set out by the indirect effects due to network inter-dependence. Analysis of the eigenvectors of matrix h (or better the stochastic distribution of them) allows to predict the most sensible component to the external disturbances. Upon disturbance release, the system recovers its original equilibrium condition. This picture represents the typical scenario of a disruptive temporary disturbance perturbing a "stable system".

When observing the same scenario at a lower level of granularity no trace of the two faulting components is left and the typical inoperability profiles do not differ from those of the non disruptive disturbance discussed in the previous session. Fig. 7 (left side) shows the typical picture.

Again the observation at lower level of granularity lacks a lot of information, in fact partial inoperability of components correspond to delays or partial delivery of the willed service; whereas complete inoperability of same component corresponds to areas where the service is not provided at all. However at the granularity of global network inoperability profiles look similar.

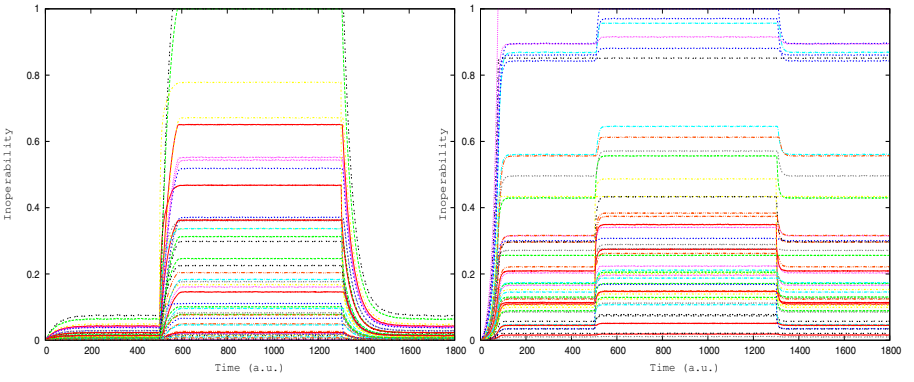


Fig. 6. All component evolution profiles of two different systems experiencing the same disturbance. Picture at the left represents a disruptive disturbance on a stable system as two different components are brought to the complete inoperability state and they are recovered on disturbance release. On the other hand, picture at the right side shows the behavior of a system shifting from the initial unstable configuration with full operability to a novel equilibrium configuration where a component is totally lost. It is worth noting that the system response to the external disturbance allows complete recovery to the novel equilibrium.

3.3 An Unstable System

The previous examples represent stable systems. This means that the propability distribution of their evolution matrices h (eq. 10) are such that no eigenvalue is greater than one. In fact, suitably low level of noise does not lead to any totally inoperable component and when the external disturbance is released the system recovers its original equilibrium. When a system exhibits an eigenvalue larger than one, some component experiences an exponential divergence up to unitariety (as cut-off applies). For those systems, any non null level of noise (regardless of its weakness) leads some component to complete inoperability. Moreover there exist components that remain always totally inoperable even when disturbance and noise are both brought to zero. This behavior fits very well our current idea of system "instability".

Fig. 6 compares the behavior of an "unstable" (right side) system with that of an other one sharing the same topology, but differing for the evolution matrix. All previously described elements of instability can be observed at the onset of the simulation. However after some hundred steps, the system achieves a novel equilibrium configuration and its response to external disturbances are such to allow recovery. It is worth stressing that, when comparing the same two systems at a low granularity level (fig. 7) no signature of the system original instability can be evidenced. All previous considerations point toward the need of some other parameter to quantify distribution of inoperability among the different

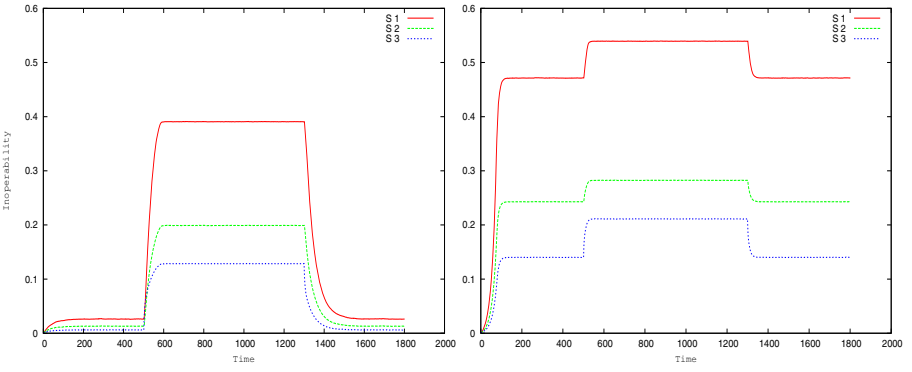


Fig. 7. The global inoperabilities of the different service networks relative to the same disruptive scenarios of fig. 6. Left side refers to a stable system experiencing a disruptive disturbance; while right side refers to an initially unstable system brought to a new stable configuration where it experiences an external disturbance.

components of a network. In this respect, several statistical indices, such as "Gini index" or "Shannon Entropy" may be useful to provide at least a part of the required information.

4 Conclusions

We have reported preliminary results of an Extended Leontief Model. Variations with respect to the Inoperability Inter-dependence Model (IIM) are introduced to improve the variety of situations to be accounted for. A stochastic Markov chain takes the place of the Leontief deterministic equilibrium equation. The inner structure of the different networks belonging to the global system is also accounted for. In this respect a "Local Dependence Approximation" is also introduced.

The paper provides evidence for the importance of the implemented scenario "granularity": different scenarios sharing the same level of inoperability at the single network level, but differing on the geographic location of the fault, evolve toward significantly different global inoperability states. Leontief original model does not suffer for such a problem, since it deals with sector products traded by several independent means; on the contrary, in the case of service network interaction, faults propagate along precise paths and the geographic position of the defective component may play a crucial role. Despite their "granularity" intrinsic dependence, we emphasize the importance of IIM's (at least as a companion tool) to provide prompt and synthetic information on the state of complex systems and their basic evolution mechanisms.

Acknowledgments. The activity on Inoperability Input/Output Models has been promoted by Sandro Bologna, who also constantly represented the relevance of such models in the perspective of potential end-users. We also benefitted of discussions with people working both in academic institutions and private companies; Emanuele Ciapessoni (CESI-Ricerca), Marino Sforna (TERNA), Roberto Setola (Campus Biomedico), Biagio Di Carlo (Telecom Italia), Jose Marti (University of British Columbia), and Erich Rome (IAIS) are kindly acknowledged. This work has been partly supported by the European project JLS/2009/CFP/CIPS/019 named "MIA" (Methodology for Interdependencies Assessment) financed by the Directorate-General Justice, Freedom and Security of the European Commission.

References

1. Watts, D.J., Strogartz, S.H.: Collective dynamics of small-world networks. Nature 393, 440 (1998)
2. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.-H.: Complex Networks: Structure and Dynamics. Physics Rep. 175, 424 (2006)
3. Leontief, W.W.: Input Output economics. Operational Research Quarterly 3(2), 30–31 (1952)
4. Jang, P., Haimes, Y.Y.: Risk Analysis 1215, 24 (2004)
5. Leontief, W.W.: The Structure of America Economy: 1919-1929 An empirical Application of Equilibrium Analisys. Harward University Press, MA (1941)
6. Santos, J.R., Haimes, Y.Y.: Risk Analysis 26(6), 1437–1451 (2004)
7. Jung, W., Santos, J., Haimes, Y.: International Trade Operability Input Output Model (IT-IIM): Theory and Application. Risk Analysis 29(1), 137–154 (2009)
8. Schneider, K., Liu, C., Paul, J.: Assessment of Interactions Between Power and Telecommunications Infrastructures. IEEE Transaction on Power Systems 21(3), 1123–1130 (2006)
9. De Porcellinis, S., Panzieri, S., Setola, R.: Modelling critical infrastructure via a mixed holistic reductionistic approach. International Journal of Critical Infrastructures 5(1-2), 86–99 (2009)

Critical Infrastructure Protection in Brazil - Threat Identification and Analysis

João H.A. Franco[1], Sérgio L. Ribeiro[1], Sandra M.C. Tome[1], Christiane M.S. Cuculo[1], Marcos B. Trindade[1], Leonardo M. Lage[1], and Regina M.F. Souza[2]

[1] Fundação CPqD – Centro de Pesquisa e Desenvolvimento em Telecomunicações
Rodovia SP-340 km 118,5 – 13086-902 Campinas, SP – Brazil
{franco,sribeiro,sandrat,ccuculo,trindade,lmlage}@cpqd.com.br
[2] Agência Nacional de Telecomunicações – Anatel
SAUS Quadra 6 Bloco E – 70070-940 Brasília, DF – Brazil
reginas@anatel.gov.br

Abstract. This paper describes the Methodology for Threat Identification and Analysis (MIdA^2) and its application to Brazil's critical telecommunication infrastructure. As this effort is part of a bigger project, Brazil's critical telecommunication infrastructure protection, other methodologies related to MIdA^2 are also briefly presented to give the reader a broader perspective.

Keywords: Critical telecommunication infrastructure protection, vulnerability identification, impact assessment, risk assessment.

1 Introduction

The complexity and the extent of modern infrastructures, their interdependencies and the need to use scarce resources in a cost-effective manner require a systematic approach to identify and protect critical infrastructures. This process should determine critical elements, identify threats and vulnerabilities, assess impacts and evaluate risks. Following these steps, actions can be taken to mitigate vulnerabilities and reduce risks. Finally, government policies and strategies can be defined.

Several methodologies that address some of these needs have been proposed: OCTAVE [1], CRAMM [2], TAME [3], among others. However, none of them fulfilled all the requirements in order to be applied to Brazil's infrastructures. To reach this goal, a set of five methodologies was developed, among them the Methodology for Threat Identification and Analysis-MIdA^2, which is described in this paper, together with its application to identify the threats in Brazil's critical telecommunication infrastructure (CTI).

The work described here has been performed together by Anatel (Brazil's telecommunication agency), and Fundação CPqD (the largest R&D institution in Latin America), with the support of Funttel, Brazil's telecommunication technology development fund.

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 14–21, 2010.

2 Methodologies for Critical Infrastructure Protection

The proposed model for Brazil's critical telecommunication infrastructure protection [4,5] is supported by a set of five interrelated methodologies[1]:

Methodology for Critical Infrastructure Identification (MI^2C) – used to identify the elements that constitute the critical infrastructure [6];
Methodology for Threat Identification and Analysis ($MIdA^2$) – used to identify threats that can affect the critical infrastructure under analysis;
Methodology for Interdependency Analysis of Critical Infrastructures (MAI^2C) – used to map the interdependencies between the critical infrastructure under consideration and other critical infrastructures previously identified;
Methodology for Ideal Scenario Creation ($M(CI)^2C$) – used to create the ideal scenario for critical infrastructure protection;
Methodology for Diagnosing Critical Infrastructure ($MeDI^2C$) – used to diagnose a section[2] of a critical infrastructure and compare the ideal and the actual scenarios in order to develop recommendations and an action plan.

Although this paper describes an application in the telecommunication sector, all five methodologies, including $MIdA^2$, can be used in other sectors such as energy, transportation, water, finance etc.

3 Critical Telecommunication Infrastructure Identification

Before $MIdA^2$ can be applied, the critical infrastructure under consideration must be identified, a task that can be accomplished with MI^2C. This methodology was first used to identify the critical telecommunication infrastructure used during the XV Pan/Parapan American Games [8], held in Brazil in 2007. Based on the experience acquired in that pilot project, MI^2C was revised and then applied to Brazil's CTI as a whole [9].

In both cases, telecommunication stations (comprising switching and/or transmission and/or wireless systems) were chosen as infrastructure elements. In the second instance, only elements that support the three most critical services identified by MI^2C were considered[3].

Besides helping to identify the elements that compose the critical infrastructure, MI^2C also allows the classification of these elements according to a criticality index. In fact, two different indexes are calculated, one related to a particular operator infrastructure (local criticality index) and the other related to Brazil's infrastructure (global criticality index). $MIdA^2$ will subsequently use both indexes.

[1] The acronyms shown are derived from the methodologies' names in Portuguese.
[2] In the sense of ISO/IEC 17799 [7].
[3] Fixed telephony, mobile and Internet backbone services.

4 MIdA^2 and Its Application to Brazil's Critical Telecommunication Infrastructure

MIdA^2 main objective is to collect and analyse threat information in order to start the risk management process, which will continue with the help of the other methodologies. MIdA^2 five phases are described below (see Figure 1) together with a description of its application to Brazil's CTI, an effort accomplished together by Anatel, CPqD and the eleven largest private telecommunication operators in Brazil.

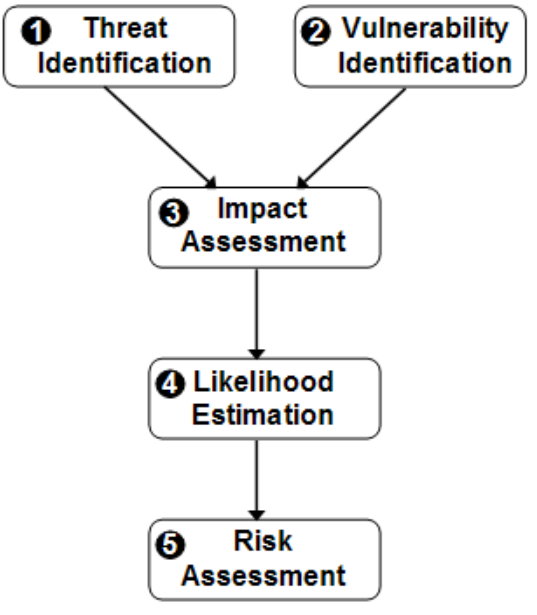


Fig. 1. MIDA^2 phases

4.1 Threat Identification (Phase 1)

Description: In this phase all potential threats that can affect the infrastructure under consideration are identified.

Application: A preliminary version of a list of threats, prepared by Anatel and CPqD, was sent to telecommunication operators for comments. The answers enabled Anatel and CPqD to consolidate a catalogue with forty six threats, encompassing flood, fire, social engineering, hardware and software failures, spying (espionage), EMI (electromagnetic interference) etc.

4.2 Vulnerability Identification (Phase 2)

Description: The focus of phase 2 is to develop a list of vulnerabilities related to the previously identified threats. Although the vulnerabilities identified here will only be used by methodologies other than MIdA^2, this phase helps reviewing the threats already recognized.

4.3 Impact Assessment (Phase 3)

Description: The purpose of this phase is twofold:

- To assess, on a qualitative basis, the expected level of service disruption (loss of availability) suffered by each element id a specific a threat occurs. The alternatives are:
  - "Severe or catastrophic service disruption";
  - "Moderate service disruption";
  - "Negligible service disruption".

  The corresponding values used in the following analysis are, respectively, 100%, 50% and 10%.
- To obtain the station recovery time if a specific threat occurs (objective information based on past history, e.g. logs). These values are measured in number of hours, assuming that all implemented controls are in place.

Application: All telecommunication operators were asked to provide this information for their most important stations according to MI^2C (i.e. the stations with the largest local criticality indexes).

4.4 Risk Assessment (Phase 4)

Description: In this phase, the following data are gathered for each element-threat pair:

- The likelihood that the threat could occur (expectation based on previous experience) – the alternatives are: [10]
  - "Virtually certain" (chance of occurrence greater than 99%);
  - "Likely" (chance of occurrence between 66% and 99%);
  - "Possible" (chance of occurrence between 33% and 66%);
  - "Unlikely" (chance of occurrence between 1% and 33%);
  - "Improbable" (chance of occurrence less than 1%).
- The frequency of occurrence of the threat (objective information based on past history, e.g. logs), measured in number of events per year (observed annualized rate of occurrence), assuming once again that all implemented controls are in place.

Application: All telecommunication operators were asked to provide this information for their most important stations according to MI^2C.

4.5 Risk Assessment (Phase 5)

Description: This analysis is performed in two steps, risk determination and risk data consolidation, detailed below.

Risk Determination. The service availability risk[4] that the threat i occurs in the station k of the operator j in the context of the telecommunication operator infrastructure, $R_L(i, j, k)$, can be expressed as follows:

[4] Other risk components, like integrity and confidentiality risks, were not taken into account in this application.

$$R_L(i,j,k) = S(i,j,k) * T(i,j,k) * C_L(j,k) * F(i,j,k) \quad (1)$$

where:

$S(i,j,k)$ – expected level of service disruption if the threat i occurs in the station k of the operator j (information obtained in phase 3);

$T(i,j,k)$ – observed recovery time if the threat i occurs in the station k of the operator j (information obtained in phase 3);

$C_L(j,k)$ – local criticality index of the station k (i.e. the criticality index of station k in the operator j infrastructure context – information provided by MI^{2}C);

$F(i,j,k)$ – observed annualized rate of occurrence of the threat i in the station k of the operator j (information obtained in phase 4).

In case $F(i,j,k)$ is not disclosed by the operator, it is replaced in expression (1) by the product $L(i,j,k) * F_{est}(i,j)$, where $L(i,j,k)$ is the likelihood[5] that the threat i will occur in the station k of the operator j and $F_{est}(i,j)$ is the estimated annualized rate of occurrence[6] of the threat i in any station of the operator j. An expected value of 10^{-1} event per year (a single occurrence every ten years) is used if the informed $F(i,j,k)$ is zero (i.e. that particular threat has never occurred). Similarly, an expected value of 10^{-2} hour (36 seconds) is used if the informed $T(i,j,k)$ is zero.

Likewise, the service availability risk that the threat i occurs in the station k of the operator j in the context of Brazil's telecommunication infrastructure, $R_G(i,j,k)$, can be calculated as:

$$R_G(i,j,k) = S(i,j,k) * T(i,j,k) * C_G(j,k) * F(i,j,k) \quad (2)$$

where $C_G(j,k)$ is the global criticality index of station k (i.e. the criticality index of station k of operator j in Brazil's telecommunication infrastructure context – information provided by MI^{2}C).

Risk Data Consolidation. The service availability risk that the threat i occurs in the context of the operator j infrastructure, $R_L(i,j)$, can be expressed as:

$$R_L(i,j) = \sum_{k=1}^{S_j} R_L(i,j,k) \quad (3)$$

where S_j is the number of telecommunication stations of operator j.

In the same fashion, the service availability risk that the threat i occurs in the context of Brazil's telecommunication infrastructure, $R_G(i)$, can be calculated in two steps:

$$R_G(i,j) = \sum_{k=1}^{S_j} R_G(i,j,k) \quad (4)$$

[5] Information obtained in phase 4.

[6] Information defined by Anatel and CPqD obtained from sources other than the telecommunication operators (news published in newspapers, risk newsletters and magazines, personal experience etc.).

$$R_G(i) = \sum_{j=1}^{N_O} R_G(i,j) \tag{5}$$

where $R_G(i,j)$ is the service availability risk that the threat i occurs in the operator j infrastructure in the global context and N_O is the number of telecommunication operators.

4.6 Summary of Results

At this point, meetings were arranged with each telecommunication operator to present and discuss the results achieved, namely:

- The service availability local risk that the threat i occurs in a particular station k of the operator J, $R_L(i,J,k)$;
- The service availability local risk that the threat i occurs in the operator J infrastructure, $R_L(i,J)$ and
- The service availability global risk that the threat i occurs in Brazil's telecommunication infrastructure, $R_G(i)$.

An illustrative example of the information presented to the telecommunication operators in those meetings is presented in the figures below. Figure 2 shows the service availability local risks for different stations of the telecommunication operator J, $R_L(i,J,k)$. Each vertical bar shows the risk percent values corresponding to 6 (six) different threats that can occur at one specific station. For instance, the threat that represents the largest risk in station B is 'software failure' (40% of the total risk), followed by 'hardware failure' (30%). Figure 3 presents some of the service availability local risks of operator J, $R_L(i,J)$, and Figure 4 shows some of the service availability global risks for Brazil's telecommunication infrastructure, $R_G(i)$.

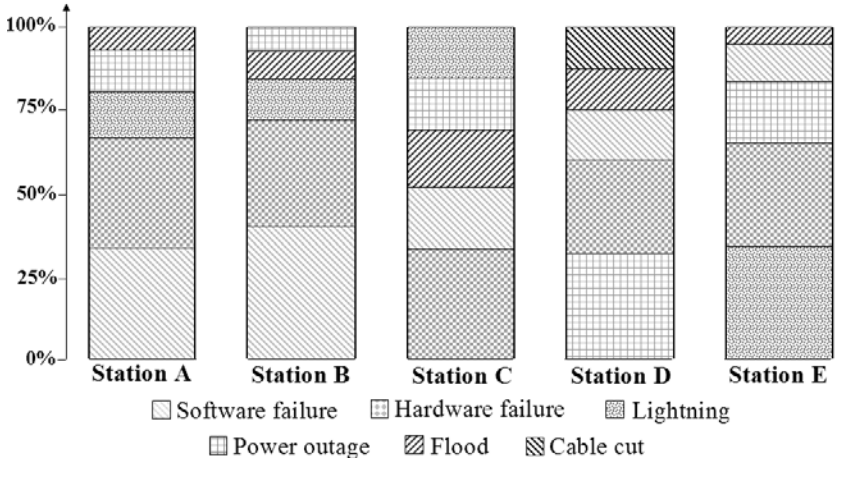


Fig. 2. Service availability risks for different stations of operator J, $R_L(i,J,k)$

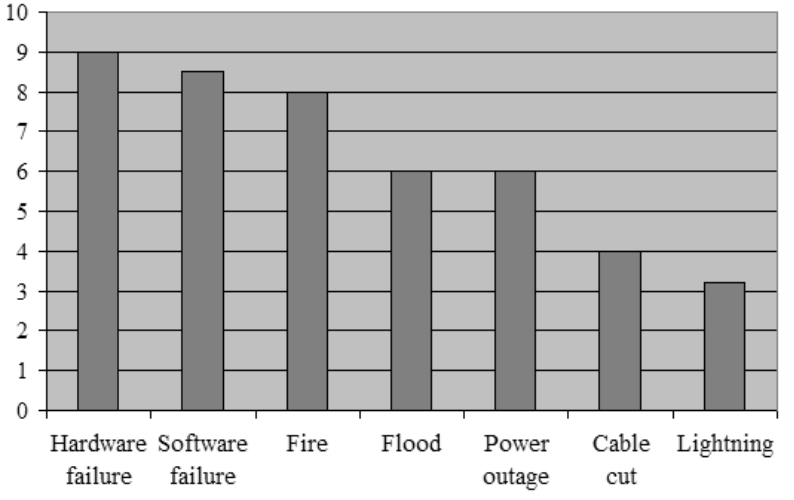


Fig. 3. Service availability risks for the operator J infrastructure, $R_L(i, J)$

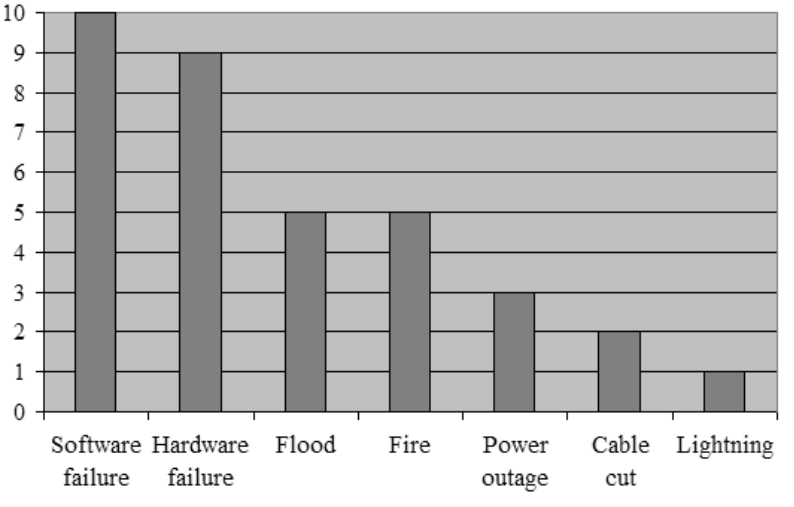


Fig. 4. Service availability risks for Brazil's telecommunication infrastructure, $R_G(i)$

5 Conclusions and Future Work

This paper described the Methodology for Threat Identification and Analysis (MIdA^2) and its application to Brazil's critical telecommunication infrastructure. MIdA^2 main objective is to collect and analyse threat information in order to start the risk management process, which will continue with the help of the other methodologies. MIdA^2 has been defined with 5 (five) phases: Threat Identification, Vulnerability Identification, Impact Assessment, Risk Assessment and Threat Analysis. The paper also presented the definition and the evaluation of

the service availability local and global risks and some illustrative examples as well.

The information provided by MIdA^2 will enable the telecommunication operators and Brazil's telecommunication agency, Anatel, to identify the threats that pose the largest risks at different levels, either organizational, geographical or national. In addition, MIdA^2 results will be used by the other methodologies with the purpose of establishing a risk management process in the context of a National Critical Infrastructure Protection Plan.

Future directions include the application of MIdA^2 to infrastructures such as energy, transportation, water, finance and others and the development of a web-based software tool with the purpose of speed up the exchanging information process among all players, Anatel, CPqD and Brazil's telecommunication operators.

References

1. Alberts, C.J., Behrens, S.G., Pethia, R.D., Wilson, W.R.: Operationally Critical Threat, Asset, and Vulnerability Evaluation SM (OCTAVESM) Framework, Version 1.0 (CMU/SEI-99-TR-017). Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA (1999)
2. CCTA Risk Analysis and Management Method-CRAMM, http://www.cramm.com
3. Vidalis, S., Blyth, A.: Understanding and Developing a Threat Assessment Model, http://www.glam.ac.uk/socschool/research/publications/technical/CS-02-3.pdf
4. Franco, J.H.A., Ribeiro, S.L.: Estratégia de proteção da infraestrutura crítica de telecomunicações. CPqD Internal Report, Campinas, Brazil (2008)
5. Souza, R.M.F.: Critical telecommunication infrastructure protection project. info@CITEL, vol. 33 (2007), http://www.citel.oas.org/newsletter/2007/marzo/infraestructura_i.asp
6. Ribeiro, S.L., Nakamura, E.T., Bezerra, E.K.: Critical Infrastructure Protection in Brazil. In: 1st IEEE International Workshop on Critical Infrastructure Protection, Darmstadt, Germany (2005)
7. ISO/IEC 17799:2005. Information technology. Code of practice for information security management. International Organization for Standardization-ISO, Geneva, Switzerland (2005)
8. Critical Telecommunication Infrastructure Protection – Application of MI^{2}C. Contribution CCP.I-TEL 1095/07. Inter-American Telecommunication Commission-CITEL, Mendoza, Argentina (2007)
9. Critical Telecommunication Infrastructure Protection in Brazil – Application of MI^{2}C. Contribution CCP.I-TEL 1448/08. Inter-American Telecommunication Commission-CITEL, Puerto Iguazu, Argentina (2008)
10. IPCC terminology about certainty, http://www.sej.org/resource/IPCC_terminology.htm

Development of Information Security-Focused Incident Prevention Measures for Critical Information Infrastructure in Japan

Hideaki Kobayashi[1], Kenji Watanabe[2], Takahito Watanabe[1], and Yukinobu Nagayasu[1]

[1] Security Engineering Laboratory, IT Security Center,
Information-technology Promotion Agency, Japan (IPA)
2-28-8, Honkomagome, Bunkyo-ku, Tokyo, 113-6591, Japan
{hd-koba,t-watana,y-nagaya}@ipa.go.jp
[2] Nagaoka University of Technology,
1603-1 Kamitomiokamachi, Nagaoka, Niigata, 940-2188, Japan
watanabe@kjs.nagaokaut.ac.jp

Abstract. In recent years, the dilemma of cyber attacks by malicious third parties targeting security vulnerabilities in information and communication systems has emerged, resulting in security incidents. This situation suggests that the establishment of proactive efforts and recurrence prevention measures are becoming imperative, especially in critical infrastructure sectors.This paper provides an analysis of 58 security incident cases, which occurred in critical infrastructures worldwide and were published in media. The purpose of the analysis is to conclude to a valid list of recurrence prevention measures that constitute good practices.

Keywords: Information security, Critical Information Infrastructure security, Security vulnerabilities, Security incidents.

1 Introduction

In the present era, Information and Communication Technology (ICT) is becoming an infrastructure that constitutes the "nervous system" of the current economy and society. However, the unauthorized access to the power grid system (United States) and water system (Australia), the prevalence of computer viruses, and the occurrence of accidents/incidents due to the system failures of financial and transportation services portray the exponential increase of the imminent danger and damage that ICT issues may instigate.

The Japanese Government established the National Information Security Center (NISC) to promote information security on April 25, 2005. NISC started "The First National Strategy on Information Security – Toward the creation of a trustworthy society –" from the fiscal year 2006 to 2008. The implementing entities were classified into four areas: a) Central Government/Local Governments, b) Critical

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 22–33, 2010.

Infrastructures, c) Businesses and d) Individuals [1]. For the Critical Infrastructures, the following 10 sectors are identified: “Telecommunications”, “Finance”, “Civil aviation”, “Railways”, “Electricity”, “Gas”, “Governmental/Administrative services (including local governments)”, “Medical services”, “Water works” and ”Logistics” [2] [3]. In the First National strategy, several policies such as information sharing (CEPTOAR: Capability for Engineering of Protection, Technical Operation, Analysis and Response), analysis of interdependency and cross-sectoral exercises etc. were promoted.

In fiscal year 2009, “The Second National Strategy on Information Security” has started and continues until the fiscal year 2011. The target scope of this Second National Strategy includes preparedness and enhancement of security measures against the incidents [4] [5].

In the midst of the current circumstances and on the premise that information security incidents will occur, it is becoming imperative for the government and private sectors to cooperate in implementing measures for incident prevention, damage control, and rapid recovery in the case security is breached.

Therefore, it is necessary to promote both security and reliability as the two wheels of a chariot to protect critical infrastructures from future ICT incidents. Recognizing information security as a risk to critical infrastructure systems, foreign and domestic incident publications were collected, the causes analyzed, and recurrence prevention measures were identified [6].

2 Analysis of Case Studies on Security Incidents in Critical Infrastructures

2.1 Approach to Incident Information Gathering

To collect information concerning security incidents experienced in critical infrastructures, two methods can be considered. The first is to gather information through publications in the media; the second is to gather information directly from critical infrastructure operators.

As for the latter, it was assumed to be particularly difficult as the details concerning security incidents are considered sensitive information by critical infrastructures. For this reason, gathering of incident information was limited to gleaning information from publications in mass media.

2.2 Subject and Range of Incidents

In gathering incident information, the time frame was limited between the year 2000 and 2008, and incidents were not limited domestically, but included foreign incidents as well. Incidents that occurred overseas were included mainly for the reason that some had great influence in relation to security (by means that were not experienced

domestically, caused much greater damage, etc.) and, by gathering and utilizing them as references, carried great weight from the perspective that they could be positively employed in preventing security incidents before occurring in our own country.

21 different domestic news sites[1] were utilized as sources for domestic incidents and 17 foreign security news sites[2] were utilized to gather foreign incident information. For this analysis, 38 domestic incidents and 20 foreign incidents were selected from a total of 58 security incident cases in critical infrastructures.

2.3 Classification of Incident Information

For the classification of the security incident cases, the items in Table 1 were extracted from the published information.

If there was a need to thoroughly collect information concerning the attacker and the method of attack, they were included in the "Incident Summary" field.

Table 1. Incident Information Items

No.	Identification number of the information security incident case.
Incident Case Title	Incident case title including main security causes and the infrastructure operator.
Date of Incident	Occurrence date of the information security incident. In the case that the occurrence date is unknown/unavailable, the date the information security incident was discovered. In the case the discovery date is also unknown/unavailable, the date the information security incident was published
Incident Summary	The summary of the information security incident should be stated. The attacker, victim party, attack method, and damage information should also be extracted and summarized concisely in several lines.
Main Causes	The excerpt of published information word for word from an article should be avoided and the causes are categorized into several types. Refer to section 2.4 for details.
Impact Range	The extent of damage generated by the occurrence of the information security incident. Particularly, numerical recording is encouraged. For example, in case of a data breach, the exact number of cases leaked should be recorded. In case of denial of service incident, the extent (number of services) and period of time services were suspended. In the case of monetary damage, the total sum of loss should be recorded.
Recurrence Prevention Measure	In accordance with the analysis summarized later, a recurrence prevention measure should be proposed, and the recurrence prevention measure ID should be recorded. Refer to section 3 for details.
Remarks	Information worth noting, such as the manner in which the information security incident was discovered and the countermeasure(s) the critical infrastructure operator implemented should be recorded.
Source	The source of the security incident information.

[1] e.g., Asahi.com (The Asahi Shimbun Company) http://www.asahi.com/,
NIKKEI NET (Nikkei Inc. / Nikkei Digital Media, Inc.) http://www.nikkei.co.jp/,
ITmedia (ITmedia, Inc.) http://www.itmedia.co.jp/

[2] e.g., DHS Daily Open Source Infrastructure Report
http://www.dhs.gov/files/programs/editorial_0542.shtm,
Industrial Defender http://www.industrialdefender.com/,
Cyber Security News http://cicentre.com/news/cyber_security.html

2.4 Analysis of Studied Incidents

In this study, security incidents experienced by existing critical infrastructures were analyzed[3]. The purpose behind gathering and analyzing information security incident cases is to identify causes and recurrence prevention measures. The method employed for analysis was the classification of information through the systematization of incident causes as shown in Fig. 1.

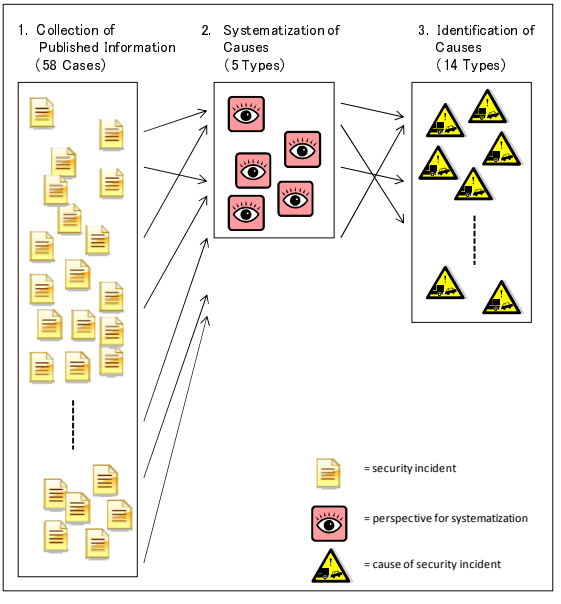


Fig. 1. Classification of Published Information

2.4.1 Cause Systematization

There were many accounts within the published information collected that could be considered as causes of the information security incidents. However, there was no consistency in the manner in which they were recorded, making it difficult to systematize the information from a single perspective. After contemplating possible perspectives from which the information could be systematized, five different perspectives were thought to be feasible. These five perspectives are explained as follows:

[3] The same method was employed by Brandstetter et al. as one of their methods. Thomas Brandstetter, Konstantin Knorr, and Ute Rosenbaum: A Structured Security Assessment Methodology for Manufactures of Critical Infrastructure Components: Proceedings of the 24th IFIP TC 11 International Information Security Conference, SEC 2009 Pafos, Cyprus, May 2009, pp248-258, Springer.

(1) Unauthorized Access

In many of the published articles, the cause was expressed as "unauthorized access". However, "unauthorized access" in short, allows for the assumption of a vast variety of different circumstances and is exceedingly vague. To overcome the generality of "unauthorized access", this group was supplemented with additional information: the attacking method and the type of perpetrators, such as "SQL Injection" (3 cases), "Denial of Service Attack by External Sources" (6 cases), "Unauthorized Access from External Sources" (10 cases), "Unauthorized Access by Former Personnel" (3 cases), and "Unauthorized Access by Internal Personnel" (2 cases). If no additional information was available, then the cause was just labeled as "Unauthorized Access" (5 cases).

(2) Inappropriate Use of Winny (File sharing software used in Japan)

There were many cases that identified Winny as the source of information leakage (8 cases), and they were systematized under "Inappropriate Use of Winny".

(3) Issues in System Development

"Inadequate Design" (5 cases), "Inappropriate Server Configuration" (2 cases), "Insufficient System Independency" (1 case), and "Network Route Falsification" (1 case) are issues encountered during the course of system development due to the lack of consideration in a certain aspect.

(4) Human Error

Causes that were brought about by error by personnel within an organization consisted of "Error by Internal Personnel" (4 cases) and "Error by Contracted Personnel" (1 case).

(5) Phishing

There were 4 information security incidents due to phishing fraud by web spoofing as a financial institution website. These were categorized under "Phishing".

2.4.2 Causes

After the systematization process using the perspectives explained in the previous section, the causes can be divided into 14 types as shown below:

(1) Unauthorized Access
   a. Denial of Service (DoS) Attack by External Sources (6 cases)
   b. SQL Injection (3 cases)
   c. Unauthorized Access by External Sources (10 cases)
   d. Unauthorized Access by Former Personnel (3 cases)
   e. Unauthorized Access by Internal Personnel (2 cases)
   f. Unauthorized Access (5 cases)

(2) Inappropriate Use of Winny (File sharing software used in Japan)
   a. Inappropriate Use of Winny (8 cases)

(3) Issues in System Development
   a. Inadequate Design (5 cases)
   b. Inappropriate Server Configuration (2 cases)
   c. Insufficient System Independency (1 case)
   d. Network Route Compromise (1 case)

(4) Human Error
   a. Error by Internal Personnel (4 cases)
   b. Error by Contracted Personnel (1 case)

(5) Phishing
   a. Phishing (4 cases)

There were 4 cases with unknown causes, and 1 case that included 2 causes.

2.4.3 Incident Analysis Table

The results of the analysis conducted in this section concerning the information security incidents above were collated as an incident analysis table. As an example, a case of SQL injection (case#52) is shown in Table 2.

Table 2. Incident Analysis Table Excerpt: Case #52

No.	52
Incident Case Title	Unauthorized Access and Website Falsification of an JOGMEC 's Public Server
Date of Incident	27 Jul 2008
Incident Summary	Japan Oil, Gas and Metals National Corporation (JOGMEC) server was compromised by SQL injection. Viewers may have contracted a virus. The computers that accessed the falsified website were automatically redirected to a server (storage site of malicious programs) set up by the attacker and malicious programs may have been downloaded forcefully.
Main Causes	SQL Injection
Impact Range	JOGMEC as well as viewers/users of the website may have contracted a virus.
Recurrence Prevention Measure	L3-P1, L2-D1, L2-D3, L2-D4, L3-O1, L3-M1
Remarks	Critical Infrastructure Sector: Governmental/Administrative services (including local governments)
Source	http://www.asahi.com/national/update/1020/TKY200810200153.html http://www.jogmec.go.jp/news/release/docs/2008/pressrelease_080918.pdf

3 Recurrence Prevention Measures Derived from Incident Cases

Recurrence prevention measures were identified and classified as seen in Fig. 2. The classification process was conducted through two perspectives – lifecycle and security level. First, recurrence prevention measures for each of 14 analyzed causes were selected (1. Causes). Then each recurrence prevention measure was categorized into one

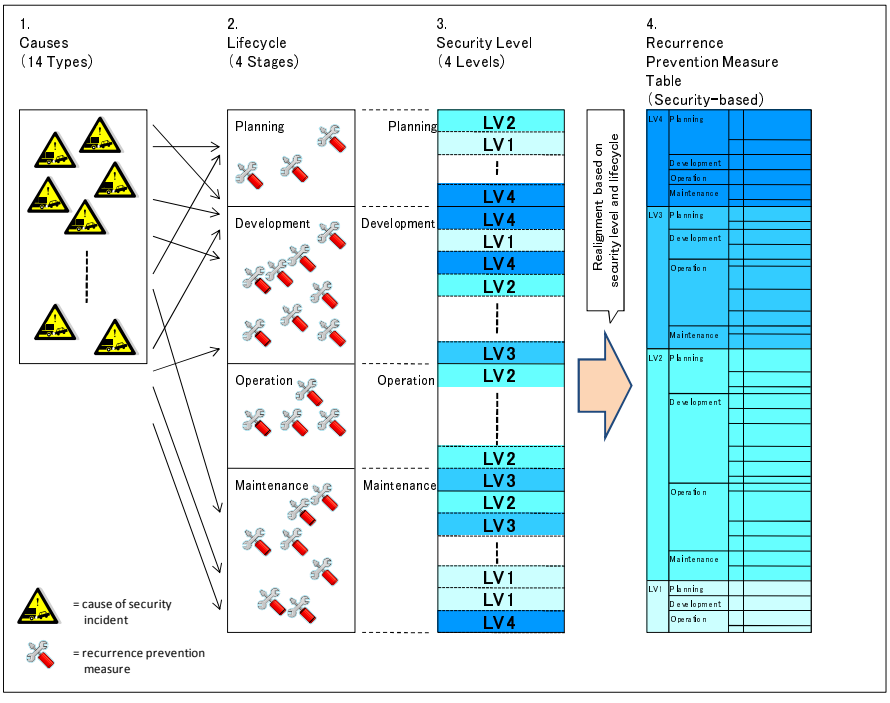


Fig. 2. Classification of Recurrence Prevention Measures

of 4 phases of system lifecycle based on its most effective phase of implementation (2. Lifecycle). Thirdly, a security level was assigned to each measure based on its criticalness (3. Security Level). Lastly, recurrence prevention measures were sorted by security levels and phases of lifecycle they are assigned, and the results from this categorization were tabulated into the recurrence prevention measure table (Appendix A) (4. Recurrence Preventions Measure Table).

3.1 Identification of Recurrence Prevention Measures

The identification process of recurrence prevention measures was conducted on the 14 different causes utilizing the following 2 methods.

1) IPA had identified the recurrence prevention measures in the light of the reliability of critical infrastructure systems [6]. These measures were assessed for any missing measures from a security viewpoint. If any, they were supplemented.
2) Taking into account the analysis results for the information security incidents in section 2, what could have been done to prevent the incident was evaluated for each incident, adding recurrence prevention measures from an information security perspective.

In this identification process, 23 recurrence prevention measures were identified and are summarized in Appendix A.

3.2 Composition of Recurrence Prevention Measures

In identifying recurrence prevention measures, the following two points were taken into consideration.

3.2.1 The Lifecycle of Information System

Information security measures are necessary at each phase of the information system lifecycle, but the overall cost can be suppressed if the measures are considered in the earliest phase possible.

Also, it is insufficient for information security measures to be conducted just once. The attackers are constantly devising new methods of attack, and new attack methods are conceived each day. For this reason, even if an information system is securely guarded at one point, it will inevitably become unsecure at a later point in time if neglected. This is why during the operation and maintenance phase of the lifecycle, it is necessary to apply recurrence prevention measures in accordance with the PDCA cycle (Plan-Do-Check-Act cycle).

Each recurrence prevention measure was categorized by considering whether the most opportune phase to apply the measure would be the "planning", "development", "operation", or "maintenance" phase.

3.2.2 Security Level

The degree of information security fulfillment in a system can be considered from a condition in which sufficient security measures are in place to a condition in which there are almost no measures applied at all. Furthermore, the necessity of security measures differs depending on the purpose of each system. For example, the implementation of virus protection software for a personal computer is the absolute minimum necessary, while that would be insufficient for backbone systems, which require periodic audits and the establishment and implementation of progressively more sophisticated measures.

After grasping that these can be divided into several stages and debated, in this document the degree of information security fulfillment is labeled as a "security level". There are four degrees of security levels, from level 4 (high level) to level 1 (low level), and each level is defined as shown in Table 3.

Table 3. Assumed Systems and Their Required Security Levels (based on system criticality)

Assumed System (Category)	Required Security Level
Backbone Information Systems in Critical Infrastructure	LV4
Backbone Information Systems in Business	LV3
Systems with Minimal Social Effect	LV2
Office and Local Systems	LV1

3.2.3 Rule of Bucket

In information security, there is a train of thought domestically called the "Rule of Bucket", which could be equivalent to the Weakest Link Principal in English. This rule implies that as water can only be retained to the lowest point that is made of the boards building up the bucket, and the security of the whole is explained in the same way. The safety offered by the information security measures is only as good as its weakest point. Consequently, even if information security is substantial in one area, the contribution it provides to the safety of the whole is minimal, implicating the necessity to create the balanced countermeasure levels at each phase of the lifecycle.

In this analysis, each individual recurrence prevention measure was considered and categorized into the four different security levels (refer to section 4.1 for details) between level 4 (high level) and level 1 (low level). If a recurrence prevention measure is said at a certain level, to satisfy that security level, the corresponding recurrence prevention measures under that level must be applied. For example, consider the level 2 recurrence prevention measures. The relevant recurrence prevention measured must be continually fulfilled to satisfy security level 2, 3 and 4.

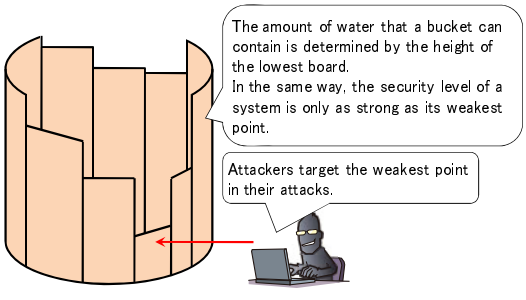


Fig. 3. Security Level Image

The recurrence prevention measure table created based on the information above is shown in the appendix below:

Appendix A – recurrence prevention measure table.

3.3 Recurrence Prevention Measure Characteristics

The measures from the recurrence prevention measure table that are characteristic from a security perspective are listed below:

(1) Establishment of Security Policy
   a. A security policy clarifies an organization's policy concerning information security.
   b. In the establishment of a security policy, it is necessary to not only consider the information system itself, but the communication and organization structure of personnel, the measures to take in case of an emergency and an security education plan also must be established in the invocation of a policy.

(2) Designing System Assuming Attack
   a. Attackers with malicious intent exist inside and outside of a system environment.
   b. Equipment failure and operation errors are assumed in the reliability context, but the assumption of attackers with malicious intent are unique to the information security perspective.

(3) Pursuit of Security in Various System Components
   a. To increase the level of security of a system in its entirety, in accordance with the Bucket Rule (the Weakest Link Principal), it is necessary to deliberate the elimination of weak sections in light of the entire system.
   b. As for OS and middleware, for example, acquisition of appropriate support contracts and implementation of appropriate measures to deal effectively with vulnerabilities that are discovered on a daily basis are necessary.

(4) Penetration Testing
   a. To close out the development phase, a security test should be performed.
   b. Apart from the systems test, non-functional testing, assuming attacks with malicious intent, are to be performed.
   c. Periodic penetration tests should be performed on the developed systems.

(5) Periodic Gathering and Check of Vulnerability Information
   a. New vulnerabilities are discovered on a daily basis and security patches are released on a daily basis even after the commencement of system operation.
   b. Planned vulnerability information acquisition and respective action should be executed for OS and middleware.

4 Self Evaluation of Security Level

4.1 Evaluation Method of Security Level

Utilizing the recurrence prevention measure table, a self evaluation can be performed to propose a measure for each security level. To be more specific, it is possible to confirm all the necessary recurrence prevention measures for the targeted security level, and requisitely for the respective security levels below it, are implemented. Every recurrence prevention measure for the security levels below must be employed to achieve the targeted security level.

To conduct a security level self evaluation, the targeted security level must be established. There are 4 different levels as shown in Table 3. Furthermore, there is no need to consider an entire corporate system as a single entity, as it is possible to separate a system into different components and to appoint each part a respective security level.

Next the items concerning the target security level and the security levels below it in the recurrence prevention measure table are utilized for the self evaluation and applied to one of the items in Table 4 to judge the current fulfillment status. For example, if level 3 is targeted, the recurrence prevention measures for level 3, level 2, and level 1 are to be checked.

Table 4. Category of Implementation Condition for Security Measures

C1:	The measure is not implemented, or it is unclear whether it is implemented.
C2:	Some measures are implemented and others are not.
C3:	All of the measures currently necessary are implemented.
C4:	The implemented measures are practically complete now and in the foreseeable future.

In a self evaluation, the recurrence prevention measures evaluated as C3 or C4 are considered to be implemented at a satisfactory level. The security level is judged as accomplished only in the event that all of the necessary recurrence prevention measures are satisfactorily implemented. This can be summarized as illustrated in Table 5.

Table 5. System Security Level Achievement Requirements

System Security Level (Category)		Criteria to Achieve Level
LV4:	Backbone Information Systems in Critical Infrastructure	All measures for LV4, LV3, LV2, and LV1 have a C3 or C4 fulfillment status.
LV3:	Backbone Information Systems in Business	All measures for LV3, LV2, and LV1 have a C3 or C4 fulfillment status.
LV2:	System with Minimal Social Effect	All measures for LV2 and LV1 have a C3 or C4 fulfillment status.
LV1:	Office and Local Systems	All measures for LV1 have a C3 or C4 fulfillment status.

5 Conclusion

The deliberation of incident countermeasures was conducted based on the analysis of published information security incidents. The recurrence prevention measures presented are some of many approaches in incident prevention, but they are derived from actual security incidents, and are thought to be sufficiently beneficial for businesses to employ as a reference in real world operation.

"The Second National Strategy on Information Security" includes promoting mitigation plans for seriously influential incidents against civil life and social economic activities in our living world where incidents will happen. Therefore this result from our study may be utilized as guideline toward the recurrence prevention measures for mitigation of IT incidents.

However, with the sophistication of attacking methods and advancement of information technology, it is necessary for countermeasures to constantly evolve as well. So it is necessary to continuously enhance our recurrence prevention measures against new threats.

From this point of view, the ways to share and analyze more detailed security incident information instead of just published reports, and develop up-to-date countermeasures based on those detailed information would be studied and improved in the future.

References

1. NISC: The First National Strategy on Information Security - Toward the creation of a trustworthy society,
http://www.nisc.go.jp/eng/pdf/national_strategy_001_eng.pdf

2. NISC: Action Plan on Information Security Measures for Critical Infrastructures http://www.nisc.go.jp/eng/pdf/actionplan_ci_eng.pdf
3. Aung, Z., and Watanabe, K., 2009, "Japan's Critical Infrastructure Protection: Risk Components and Modeling Framework" in IFIP WG 11.10 International Federation for Information Processing, Volume xxx, Critical Infrastructure Protection III, (Boston: Springer). (in printing)
4. NISC Japanese Government's Efforts to Address Information Security Issues (November 2007) http://www.nisc.go.jp/eng/pdf/overview_eng.pdf
5. NISC The Second National Strategy on Information Security (currently Japanese only) http://www.nisc.go.jp/active/kihon/pdf/bpc02_ts.pdf
6. IPA: Report of study committee for reliability of critical infrastructure information systems http://sec.ipa.go.jp/reports/20090409.html

Appendix A

	Phase	Measure ID	Details
LV4	Planning	L4-P1	Infrastructure (network, server, etc.) that can withstand intentional attack from an external source (DDoS: Distributed Denial of Service attacks, etc.) are considered and established.
	Operation	L4-O1	A measure is installed and maintained in which only software approved by authorized personnel can be installed.
		L4-O2	Sufficient security education is applied to personnel and drills have been implemented that are in line with security incident response plans (personnel contact information, media communication, etc.).
LV3	Planning	L3-P1	Security policy is implemented based on the presumption of attack from external sources.
		L3-P2	Risk analysis has been conducted presuming an event in which internal users conduct a malicious act on the system.
		L3-P3	If a section of the system has been intruded, an implemented function is distinguished and allows for the isolation of the problem so that other components are not affected.
		L3-P4	A system to respond to security incidents (CSIRT: Computer Security Incident Response Team) exists.
		L3-P5	Security logs that take forensic use into consideration are considered.
		L3-P6	A system has been established that sufficiently manages the outsources.
		L3-P7	When outsourcing, security requirements are also included in the specifications.
	Development	L3-D1	A mechanism to store security logs is equipped.
	Operation	L3-O1	Real-time monitoring (firewall, IDS: Intrusion Detection System/IPS: Intrusion Prevention System, WAF: Web Application Firewall, security logs, etc.) is conducted for security attacks such as unauthorized access.
		L3-O2	Operators are given only the minimum level of authorization to access necessary information, and a mechanism is implemented that does not allow the access to unnecessary information. Furthermore, the prompt update of authorization is conducted in the event of relocation of operators.
		L3-O3	Penetration tests on servers and web applications are conducted periodically.
	Maintenance	L3-M1	Penetration tests (tests for vulnerability such as cross-site scripting and SQL injection) have been conducted on web applications in maintenance phase.
LV2	Planning	L2-P1	In addition to availability and integrity, risk analysis concerning confidentiality is conducted.
	Development	L2-D1	A well thought out system is structured with security (firewall, IDS: Intrusion Detection System/IPS: Intrusion Prevention System, WAF: Web Application Firewall, etc.) taken into consideration.
		L2-D2	Specification reviews are conducted from a security perspective (unauthorized access, viruses, phishing, etc.).
		L2-D3	Penetration tests are conducted on servers.
		L2-D4	Penetration tests (tests for vulnerabilities such as cross-site scripting and SQL injection) are conducted on web applications in development phase.
	Operation	L2-O1	Security vulnerability related information is gathered and analyzed periodically, and measures have been taken. Security files, such as virus definition files of antivirus software, are updated appropriately.
		L2-O2	Security education, taking phishing into consideration, is conducted for users.
LV1	Development	L1-D1	Security review is conducted for source codes.

Design of a Mobile Agent-Based Adaptive Communication Middleware for Federations of Critical Infrastructure Simulations

Gökçe Görbil and Erol Gelenbe

Imperial College London
Department of Electrical and Electronic Engineering
Intelligent Systems and Networks Group
SW7 2BT, London, UK
{g.gorbil,e.gelenbe}@imperial.ac.uk

Abstract. The simulation of critical infrastructures (CI) can involve the use of diverse domain specific simulators that run on geographically distant sites. These diverse simulators must then be coordinated to run concurrently in order to evaluate the performance of critical infrastructures which influence each other, especially in emergency or resource-critical situations. We therefore describe the design of an adaptive communication middleware that provides reliable and real-time one-to-one and group communications for federations of CI simulators over a wide-area network (WAN). The proposed middleware is composed of mobile agent-based peer-to-peer (P2P) overlays, called virtual networks (VNets), to enable resilient, adaptive and real-time communications over unreliable and dynamic physical networks (PNets). The autonomous software agents comprising the communication middleware monitor their performance and the underlying PNet, and dynamically adapt the P2P overlay and migrate over the PNet in order to optimize communications according to the requirements of the federation and the current conditions of the PNet. Reliable communications is provided via redundancy within the communication middleware and intelligent migration of agents over the PNet. The proposed middleware integrates security methods in order to protect the communication infrastructure against attacks and provide privacy and anonymity to the participants of the federation. Experiments with an initial version of the communication middleware over a real-life networking testbed show that promising improvements can be obtained for unicast and group communications via the agent migration capability of our middleware.

Keywords: Communication middleware, mobile agents, peer-to-peer overlays, critical infrastructures, simulation federation.

1 Introduction

Critical infrastructures (CIs) are systems that are essential for the normal operation of society and of the economy; electricity and power generation and

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 34–49, 2010.

distribution, telecommunications, transportation, and water supply and distribution systems are some of the assets that fall under the CI categorization. A disruption of any such system will have significant adverse effects, not only on the immediately affected infrastructure and its resident region or country but also to other infrastructures and regions that are dependent on the affected systems. Thus governments, owners and operators of CIs are especially interested in critical infrastructure protection (CIP), which relates to the preparedness and response to incidents that affect the normal functioning of CIs within a country or region. This increasing concern for CIP is highlighted by the recently adopted European Programme for Critical Infrastructure Protection (EPCIP), which was created to identify and protect CIs that, in case of fault, incident or attack, could seriously impact its hosting country and possibly at least one other European member state.

The difficulty of CIP research is exacerbated by the fact that many CIs are inter-dependent, both in terms of sectors and regions, meaning a fault in a CI in one sector (e.g. power) will affect CIs in other sectors within the same region (e.g. transportation and telecommunications) and possibly also CIs in other regions. Many of these dependencies may not explicitly be known beforehand and must be discovered through research and experimentation. The difficulty in conducting CI research lies in the fact that existing systems cannot be used or compromised for experimentation, and building and maintaining real-life CI systems merely for experimentation is time-consuming and extremely costly. These difficulties are especially apparent in CIP research which deals with multiple and interdependent CI systems. Thus a good solution to this problem is the use of simulation [1], which obviously will introduce other complications and challenges.

Current CI simulators tend to be built as monolithic and highly specialized software, and the development of a single simulator combining multiple CI sectors is a complex and expensive task. This approach also fails to reuse current software and expertise. Thus in order to evaluate the interconnected behaviour of systems of diverse CIs, a good approach is to use existing CI simulators in a distributed fashion rather than to develop a single simulation system for the aggregate, so that multiple simulators will coordinate and cooperate. Since each CI simulator is typically quite costly and contains domain specific knowledge, such simulators will often be proprietary, and will be stored, managed or updated and run in geographically distributed and distinct centres. Access to distinct simulators is then only available via a wide-area network (WAN), typically the Internet. Users who would like to organise and run a joint simulation would then need to form a federation of simulators for the simulation of multiple CI sectors.

In a federated simulation system, a group of modeling and simulation tools residing on different computers are networked over a WAN and jointly operate as a federation, collaborating and coordinating with each other to accomplish a common simulation task. This allows the reuse of existing simulation models and tools and reduces the cost of developing and maintaining simulation models. The distributed approach to CI simulation for the purposes of CIP is gaining

popularity [2,3] and the EU/FP7 research project DIESIS[1] [4] provides an instance for investigating this approach. DIESIS proposes the establishment of a European modeling and simulation e-infrastructure based upon open standards to support research on all aspects of CIs with a specific focus on their protection.

Thus, the work presented here is part of the research conducted by the Imperial College team in DIESIS, where we design the core of this federated simulation e-infrastructure as a middleware that enables interoperability between CI simulators and federations of CI simulations through the Internet.

In this paper, we present the design of the communications layer, which is part of the e-Infrastructure Interoperability Middleware (EIIM) of DIESIS project, depicted in Fig. 1. The EIIM provides components necessary to interoperate different simulation tools within a federation, such as simulation time synchronization, federation management and control, scenario configuration, a knowledge base system (KBS) together with ontologies, grouped into the federation logic layer which can be accessed by the federate through the federate/middleware interface. The bottom layer of the EIIM consists of the set of network communication facilities grouped into a communications layer (CL). Central to the design of the CL is the use of communication elements (CEs), which provide adaptive communication services [5] to federates and the middleware itself for the effective and QoS-aware [6,7,8] exchange of control and information messages.

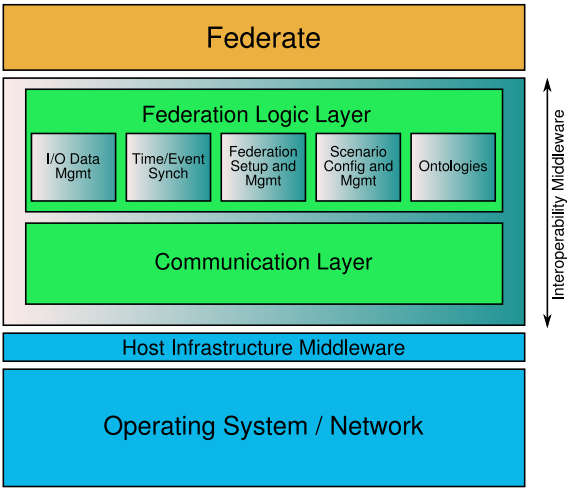


Fig. 1. The e-infrastructure interoperability middleware (EIIM)

The rest of this paper is organized as follows: Section 2 provides a brief overview of existing technologies and current research in the field of simulation federation. Section 3 presents the operational setting and requirements in

[1] DIESIS stands for "Design of an Interoperable European federated Simulation network for critical InfraStructures".

respect to federation communications. We present the design of the architecture for the CL in section 4 and discuss security issues in federation communications in Section 5. Experimental results are presented in Section 6; we conclude and discuss future work in Section 7.

2 Federating Simulations

Several standards have been proposed and used, mainly in the military domain, in order to enable distributed simulations and federations over communication networks. These standards include Distributed Interactive Simulation (DIS), Aggregate Level Simulation Protocol (ALSP), and the well known High Level Architecture (HLA).

DIS [9] is an open standard for conducting real-time platform-level wargaming across multiple host computers and is the basis for human-in-the-loop military training systems, such as the Combined Arms tactical Trainer (CATT). DIS provides no time coordination between independently running real-time simulators; interaction between simulations in DIS federations is achieved by the broadcasting of Protocol Data Units (PDUs), which are broadcast using the unreliable UDP protocol over a best-effort IP network, and therefore may be lost during transmission. Since DIS does not provide a time synchronization mechanism or communication facility other than broadcast over UDP, it is not suitable for federations which require explicit time management, message delivery guarantees and reliability.

ALSP [10] is a protocol and supporting software used extensively by the US military to interoperate analytic and training simulations; like DIS, it was superseded by HLA. Unlike the real-time simulators that participate in DIS simulations, ALSP caters for discrete-event simulators that explicitly manage their time. ALSP provides time management services to coordinate simulation times and preserve event causality across simulations. However, ALSP does not provide any support for real-time communications and although federation communications in ALSP are reliable, they suffer from performance issues, inhibiting the federation from proceeding in an efficient manner, especially in federations with participants dispersed over a wide area.

HLA [11] is a general-purpose architecture that enables distributed computer simulation systems. It seeks to generalize and build upon the results of DIS and ALSP and consists of three parts: (1) compliance rules, governing certain characteristics of HLA-compliant simulators, (2) object model templates, which define a basis for the exchange of data and events between simulators, and (3) run time infrastructure (RTI), which is a collection of software that provides commonly required services to simulation systems. The HLA RTI [12] acts as a distributed operating system for the federation, providing sufficient time management functions so that real-time, time-stepped, event-driven, and optimistic time warp simulations can all run in the same federation [13]. HLA supports these capabilities provided that federates adhere to certain rules that are necessary to realise each service. In order to participate in an HLA federation, a

federate needs to implement the HLA rules, which define the responsibilities of each federate and of the federation. The HLA RTI then provides the services required by federates during federation execution, such as communication and time management services.

HLA aims to enable various types of federations and facilitates for real-time and as-fast-as-possible (AFAP) execution of distributed simulations. However, there are almost no HLA-compliant CI simulators. Furthermore practical experience with the HLA RTI shows that it is not suited for federating highly dynamic simulation systems [14]. Since communication services offered by HLA RTI are not adaptive, they are not suitable for operation in highly dynamic and failure-prone environments. HLA RTI's support for real-time communications is also minimal and insufficient, especially in stress conditions. These shortcomings of HLA have been recognized and partially addressed, either as proposed add-ons or modifications to the RTI [15,16,17,18] or as completely new frameworks [19,20]. Unfortunately, these proposals have not yet become part of the standard. Moreover, most of the proposals fail to adequately address at least one of the communication requirements of reliability, security, and efficiency (e.g. QoS guarantees) by either restricting their simulation types of interest (e.g. by considering only analytic federations of discrete-event simulations) or the operating conditions that are assumed about the network, e.g. a high-speed local network with no failures.

In order to support reliable and real-time federation communications over dynamic and failure-prone networks, the communication infrastructure must adapt to changes in the physical network, it must be fault-tolerant and efficient. To decrease the need for administrative support, including manual setup and configuration, the communication infrastructure should also be self-configuring. For particular federations, data privacy and network security are also important requirements. Since these requirements are particularly important when the simulations are being used in preparation or in response to a real and critical incident which may be a result of natural or man-made causes, our proposed CL, described in section 4, aims to provide a communication architecture that will meet these important challenges.

3 Requirements of Federation Communications

The EIIM, our interoperability middleware [21] for federations must allow each individual federate to freely join and leave a federation without full knowledge of or by its peer federates. It should allow quick and easy assembly of independently developed simulation models and offer maximum independence to each simulation model. The user interested in the results of the joint simulation should be shielded from how or where these tasks are executed and also from the implementation details of the simulators and of the CL middleware. The communication infrastructure must also enable federates to join and leave the system without interruption to the operation of ongoing federations, and provide mechanisms for the discovery of services offered within the system.

EIIM aims to interconnect a wide variety of modeling and simulation tools into a heterogeneous federation, where federates may differ in their simulation domain, execution environment, and time management mechanisms. A federate may be implemented in one or more physical machines (e.g. with a computing cluster or grid). Various federation types are supported, not only in the nature of the participating federates (i.e. homogeneous vs. heterogeneous federation) but also in the purposes of the federation.

Distributed virtual environments (DVEs) aim to create networked interactive environments, such as virtual worlds, which are useful for training or the observation of the simulated systems in real-time. DVEs typically execute in actual or scaled real-time with a wallclock-driven simulation progress and place strict real-time constraints on the communication middleware. DVEs may tolerate some message loss but they have austere message delivery deadlines for federation messages. While DVEs require QoS guarantees, analytic simulations and decision tools, characterized by the AFAP execution pace model, require reliable delivery of messages. Low communication latency is not strictly required but the federation would not perform well if latency is high. Correspondingly, the communication middleware needs to concurrently accommodate different, and possibly conflicting, communication needs, types of federates and federations.

In order to meet these different requirements, the CL provides multiple communication services, such as reliable and real-time message delivery, under dynamic federation and network conditions. To achieve these objectives, the CL must be flexible and adaptive and self-aware [22,8]. In order to increase usability and applicability, the communication middleware must run using standard network protocols. The use of the Internet and the IP protocol stack as the main communication means for federations offers great flexibility to reach a large number of participants. However, it also creates challenges to the design of the communications architecture because the Internet essentially offers a *best-effort service* that is generally unsuitable for most federation communications.

Thus, in the next section, we propose an intelligent and adaptive communication middleware solution that provides reliable and real-time one-to-one and group communications for heterogeneous federations running over a best-effort IP network, and specifically over the Internet.

4 Architecture of the Communication Middleware

In order to address the communication requirements discussed in section 3, we propose an agent-based, intelligent, adaptive and resilient communication middleware, to provision the required smart, reliable and real-time communication services. The middleware consists of multiple **virtual overlay networks (VNets)** running over a *physical communication network (PNet)*. Each VNet is a *peer-to-peer* (P2P) network of *autonomous, mobile, goal-based software agents*, called **communication elements (CEs)**. Each VNet offers a set of communication services to federates and the EIIM, and all communications are assured by the CEs. This overlay approach does not require any modifications to the

PNet components or protocols, and all the logical functionality is embedded in the CEs which use the capabilities of the physical nodes (PNs) to achieve their goals. The proposed solution based on the CL provides the following capabilities:

- Seamless and reliable communication services in dynamic and unreliable environments: federates can communicate even when the underlying PNs are mobile and subject to failures.
- Resilience to volatility in the underlying network infrastructure: by adopting a self-organizing and distributed P2P architecture, VNets are able to operate even if a subset of the PNs is unavailable due to failures or other incidents.
- Decentralized service discovery: networked resources such as the available simulators, simulation models and data, are discovered in a distributed fashion over the P2P architecture, without reliance on a centralized directory facilitator.
- Self-monitoring and real-time service provisioning: each CE proactively monitors its own performance and if it observes that it cannot fulfill its users' communication requirements according to desired levels on its current host, then it may decide to move to another host where it may perform better. This decision is taken autonomously and CE mobility is transparent to the federates.

We assume that the operating environment of the system is dynamic and volatile, with potential failures in the PNs and the links in the PNet, and possible network disconnects because of mobility of PNs. Network load, communication patterns and requirements of federates may change at any time, as well as the set of PNs and federates in the system. All components within the system can be heterogeneous and the PNet may consist of different types of nodes, e.g. with different capabilities and operating platforms, and different types of networks. The layered approach has the advantage of insulating users of the communication middleware from the dynamic changes that occur in the environment, with the VNets providing all the access functions. VNets have no control over the mobility of PNs, and the CEs optimize VNet services through migration over the PNet and dynamic adaptation of the P2P overlay. Figure 2 presents an example federation and the system components, which are described in the rest of this section.

One major consideration for the communication middleware is portability over different operating systems and platforms. Thus the CL favours the use of services from a *host infrastructure middleware* (as presented in Figure 1) to access network and inter-process communication (IPC) facilities of the underlying operating system. Higher-level standard middleware services are avoided (i.e. distribution middleware), because federated communications are mainly message oriented and as such, they have limited need for remote functions or for direct access to remote objects. The components within the CL communicate asynchronously. IPC, which is much more efficient and faster than network communication, is employed when the communicating entities are resident on the same PN. Otherwise, network communication is used.

The federate is distinct from the middleware, and it is the user of all services provided by the EIIM. A PN may host several federates and a simulator may

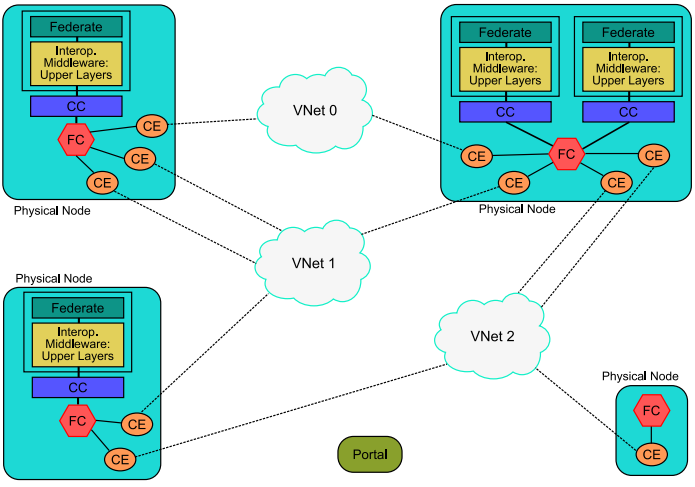


Fig. 2. A federation of simulators, connected through VNets

participate in multiple federations at the same time. Each participation of a simulator is considered as a separate federate by the EIIM. When a federate is instantiated at a PN, a local communication controller (CC) is created to handle all its communication requests. The CC is always located at the same PN as the federate and acts as the intermediary between the federate and the CEs.

Communication Controller. The communication controller acts as the communication endpoint for its associated federate. All communication requests of a federate are first handled by its CC, which selects the best VNet (and therefore CE) to use for the requested communication type. VNets that have been created to handle the communications of a federation are always registered at the CCs of the federates, either at CC or VNet creation time. For each type of VNet, a CE is registered with each CC to handle the communications of the corresponding federate. Since registered CEs may be local or remote, the CE list within a CC includes the current location (e.g. address of the host PN) of each CE and this list is updated by the migration management module of the federation controller (FC) when CE migrations occur. Once the CE that will handle the communication request is determined by the CC, the request is passed on to that CE, using IPC when the CE and CC are co-located, or via the network when the CE is remote.

When messages destined for the federate arrive, they are processed according to their priorities and types by the CC. Message processing includes ordering of messages (e.g. for timestamp-order message reception) and scheduling to prioritize messages (e.g. give processing precedence to real-time messages with closer deadlines), which are done within the CC. The dropping of messages, which have missed their deadlines and therefore are of no further use to the federate, is also done within this component.

Communication Elements. Each communication element (CE) is an autonomous, mobile, goal-based software agent that resides on a PN at any time. PNs provide physical communication and computation capabilities to CEs and a CE uses the PN it currently resides for all its physical needs. A PN may host different CEs during its lifetime and it may host multiple CEs, as well as multiple federates, concurrently. The PNet handles the physical routing of messages between nodes and we assume that the PNet provides a best-effort communication service.

CEs form a distributed P2P overlay network on top of the PNet and each CE (peer) provides one or more communication services. The goals of a CE may be dictated by the VNet it belongs to, and CEs operate and migrate in order to accomplish their goals, where these goals are quality-of-service (QoS) requirements such as minimum end-to-end latency, real-time delivery within a deadline, etc. The mobility of CEs is autonomous and they move to explore better options for their activities. A CE acts as a router for other CEs in the system, forwarding messages on behalf of others. The CL consists of multiple VNets, each VNet providing a different set of communication services to federates based on QoS criteria. While CEs within the same VNet interact and collaborate with each other in order to provide communication services, CEs across different VNets do not normally interact with each other.

The CL operates through VNets, which are self-organizing P2P overlays, and therefore knowledge of the underlying PNet is normally not explicitly needed by the CL. However, CEs continuously monitor their own performance and adapt their behaviour (e.g. the multicast tree or their host PN) in order to offer smart and efficient communication services in dynamic and failure-prone networks. While some types of CEs will be stationary, other CEs may migrate several times during their lifetimes for maintenance, resilience, reliability, security or optimization purposes.

Each CE monitors its performance continuously, and when it detects a significant decline in QoS such as an increase of end-to-end message latency, it probes its peers in the VNet and/or the PNet (based on the performance metric of the CE) in order to learn the current state of the PNet and identify candidate hosts from where it could provide better service. Once a better PN is identified, the CE signals the FC of its host and requests migration. The local FC contacts the remote FC and if the CE's migration request is accepted, then the CE's jobs and services are stopped, it is serialized, and transmitted over the network to its new PN. The FC at the recipient site creates the CE object from the received stream and starts the CE's jobs and services. The initiating FC is also responsible for buffering messages that may arrive for the CE during its migration and ensuring that the CE gets these messages after migration is complete.

Within the CL, each CE is implemented as one or more threads controlled by the CE itself. In order to enable broad applicability and ease of modification, a modular and extensible approach was taken for the design of CEs: CE behaviour is composed of *jobs*, which are reusable agent behaviours and a CE may combine multiple jobs to compose its overall behaviour. Jobs are owned by the CE and the

owner CE has complete control over their execution. Each job runs in a single thread and job threads are protected from outside access (e.g. a CE cannot manipulate another agent through controlling its job threads). A job has access to the internal structures of its CE and can communicate with its CE and other jobs in the CE through asynchronous message passing, using IPC facilities.

Federation Controller. The federation controller (FC) is an operating system resident process (e.g. a UNIX daemon or Microsoft Windows service) that manages the participation of a PN in the PNet and the EIIM. Only one FC runs per PN and the FC on a PN enables the execution and migration of VNs, acting as the "agent world". The FC provides access to the PN's communication facilities for the CCs and CEs on the same PN and implements the host infrastructure middleware to improve portability. The FC provides at least three communication sockets: UDP, TCP, and reliable UDP (R-UDP). Additional sockets may be provided depending on the PN's capabilities. Each type of socket is controlled by a *communication module*. Addition of other sockets as required by the federation and available through the PNet can easily be accomplished due to the modular design of the FC: since all logical functionality is grouped in other components, a new socket may be provided by the addition of a new communication module to the FC. The FC keeps track of the CCs and CEs on its PN in the CC and CE lists. Note that the CE list within the FC is different from the CE list within the CCs, as the FC's CE list contains all CEs on its PN whereas a CC's CE list contains all CEs (local and remote) that have been associated with that CC and corresponding federate.

The FC contains the *network statistics repository*, which is a common database that is shared by CCs and CEs and contains observations of the network, such as measured end-to-end delays to peer PNs. The purpose of this repository is to offer an up-to-date information about the network to CEs to allow them make better configuration decisions (e.g. for migration). The database will typically be updated by CEs as they continuously use the network and evaluate its performance. The location of the database is in the FC since network observations are specific to the PN where the observations are made. The agent management module within the FC manages the execution of the CEs. Since CEs operate autonomously, the FC does not control CE execution by itself but rather according to the requests made by the CEs themselves. The migration management module controls CE migration to and from this PN.

The *message dispatcher* forms a central part of the FC, handling all message communications from/to the local CCs and CEs and de-multiplexing messages to their intended destinations. The message dispatcher uses the CC and CE lists and the message parameters to determine if a message's destination is local, i.e. on the same PN. If the message is local, then IPC is used to forward the message to its destination; otherwise, one of the communication modules (i.e. sockets) is chosen according to the message type to dispatch it through the network. The message dispatcher may incorporate a temporary queue and a scheduler to prioritize among messages.

5 Security

Security is an important aspect of communications, particularly when data confidentiality is required between participants in a federation. In order to provide secure communications, the CL can incorporate standard security methods, such as encryption/decryption, authentication, verified signatures, etc. The design of the CL is modular due to the use of VNets and additional VNets employing standard security methods may be created that provide secure communications between federates. In addition to the three standard sockets provided, the FC may also offer access to native secure sockets via the PNet. When native secure sockets are not available from the network, secure communications may be provided by the use of SSL or TLS over IP. CEs (and potentially federates) may migrate in order to increase security within the network. For example, a CE may migrate from an insecure PN to one that is more secure (e.g. because it hosts native security methods), thereby increasing the security of its communication services.

The CL uses mobile agents to provide adaptive and efficient communication services. It should be noted that our middleware only supports agents (i.e. CEs) developed as part of the CL, and agents developed by other parties cannot migrate and execute on the PNs. Therefore, there are no security issues related to accepting and executing foreign and unknown mobile software. However, there is still the possibility of malicious hosts (i.e. PNs) trying to influence and affect other PNs by corrupting their resident mobile agents. The agents within the CL are designed so that they operate autonomously, so no other entities other than themselves can change their data or execution state. A malicious PN can still block all agent operations on itself, which would be synonymous to that PN having a failure, in which case other agents would ignore that PN and adapt their operations according to the observed failure.

We also need to think about threats that come directly from the network, such as from compromised hosts and denial-of-service (DoS) attacks which are a common form of cyber-warfare and a threat to network security. A typical DoS attack is often distributed (DDos), where the attacker takes control of a number of lightly-protected or compromised computers and uses them to send a large number of packets to the victim PNs. The PNs and links in the vicinity of the target(s) become overloaded and legitimate clients are unable to access or use the targeted PN(s).

DoS defence requires the detection of the attack via the classification of network traffic as normal and DoS, and response to the attack [23,24]. The first step requires fast detection before destructive traffic actually builds up. In [25], a detection scheme using Bayesian classifiers and random neural networks (RNN) is proposed. As a first step, the features to be used in detection are selected. Then, normal and attack traffic patterns are used to train the RNN to discriminate between two types of traffic. In the detection phase, a decision is given about the category of the incoming traffic (i.e. normal or attack) using the trained RNN. Using these approaches, an integrated defence mechanism against DoS attacks, incorporating the Bayesian classifier-based detection mechanism with response approaches of prioritization and rate-limiting, will be developed using the

approach in [25]. CEs may also integrate several different DoS response methods for each aspect of DoS defence.

6 Experimental Results

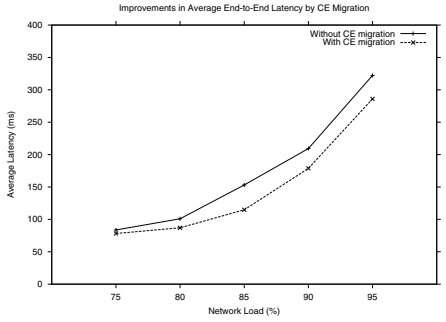
An initial version of the proposed communication middleware has been implemented in C++ with a multi-thread model for efficiency and decreased response time in operations. We have elected to use the Boost C++ libraries [26] to provide the host infrastructure middleware facilities. The network testbed we use consists of 16 physical routers connected in a static real-life topology, running IPv4 and using least-hop routing at the physical layer.

In order to emulate realistic traffic that may be generated by CI simulations, we used an existing wireless mobile ad hoc network simulation on the OMNeT++ discrete-event network simulator [27] and stored all the simulation events in a file. A federation of simulators was emulated by sub-dividing the network's simulated area into four equal quadrants and a federate was then associated with each quadrant[2]. The interactions that take place between these four areas then correspond to interactions between the federates that are installed in the testbed nodes. Since the communication protocol stack is simulated down to the link layer (i.e. MAC protocol level), the federates generate a significant amount of network traffic. Due to the broadcast nature of wireless communications simulated in OMNeT++, federates create both one-to-one and one-to-many communications, which are handled by the communication middleware.

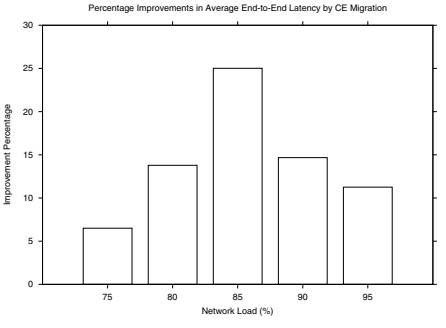
Each data point in the presented results consist of five runs of a federation, where the four federates were placed randomly within the physical network. The bandwidths of the physical links were artificially limited and for each run, random cross-traffic was introduced into the network to create dynamic network conditions. We then evaluated the performance of the communication middleware, with and without agent migration, under different network load conditions. The performance parameter of interest was selected as end-to-end message delivery latency; in the case of one-to-many communications, each message was considered separately. After an initial setup phase of 5 seconds, a federation was executed until simulation completion (about 120 seconds), during which cross-traffic patterns in the physical network were changed every 10 seconds.

Figure 3 summarises the results of these experiments. We observe that CE migration reduces message delivery latency under all network load conditions due to the smart repositioning of CEs over the PNet. CE migration adapts to changes in the PNet, and CEs relocate themselves when traffic conditions change,

[2] This is an example of a single-domain federation where only a single type of simulator is used. While the fidelity of this example application can be considered low when we consider all aspects of CI federation, including federation control and management, this paper focuses on the design of the CL, which does not use the actual data created by different types of federates for its operation but instead uses the delivery requirements for such data. Therefore, our example application is suitable to observe the performance of the CL as it covers all delivery requirements for federates.



(a) Latency values



(b) Percentage improvement

Fig. 3. Improvements in average end-to-end latency by CE migration

as shown in the experiments. Improvements in end-to-end latency with CE migration range from 6% to 25% as shown in Fig. 3. CE migration is especially effective when the network is moderately loaded (e.g. 80%-85%); under lighter load, CE migration does not provide a significant improvement as the PNet can route messages with low latency without the need for dynamic adaptation. CE migration is also effective under heavy load (90%-95%), but we observe a diminished improvement due to the fact that although CEs migrate to better locations over the PNet, the whole network is heavily loaded, increasing optimal latencies.

It is worth noting here that the current middleware does not incorporate adaptive overlay methods for efficient group communications, such as the creation and use of application-layer multicast trees which combine dynamic overlay adaptation (i.e. changing connections between tree nodes) and CE migration. We expect that the addition of these methods will improve group communication efficiency greatly.

7 Conclusions and Future Work

This paper has shown that in order to provide highly reliable, real-time and smart communication services for the federation of CI simulators, the communication

infrastructure needs to address the mobility, unreliability, protection/defence, and dynamic aspects of physical networks. In order to offer smart, reliable, and real-time services to users in mobile, dynamic, and unreliable environments, we propose the use of a mobile agent-based, adaptive and resilient communication middleware. Our proposed middleware consists of virtual overlay networks (VNets) running over the physical communication network (PNet), and each VNet is a self-organizing P2P network of autonomous, mobile, goal-based software agents, called communication elements (CEs). Each CE offers a set of communication services to federates and all communication functions are assured by the CEs. VNets adapt to changing conditions in the PNet through self-monitoring, communication, and migration of their component CEs. The VNet approach allows the easy addition of different services to the system, such as group and secure communications.

Our preliminary experiments with the proposed communication middleware on a real-life networking test-bed, which enables us to test the performance of the VNet approach in real-life situations using physical machines with different capabilities and operating systems, show that agent migration is a viable option to increase communication efficiency of CI federations. Our results indicate that CE migration improves end-to-end latency of message delivery for both one-to-one and group communications. The next step in the development of the middleware is the full implementation of communication modules and the inclusion of reliable and adaptive group communications, such as application-layer multicast protocols, in order to improve group communications for federations of CI simulations. We will also conduct experiments using a heterogeneous and dynamic network with failures to investigate the effects of network failures on the communication and coordination of CI federations using EIIM in a networked environment.

References

1. Rinaldi, S.M.: Modeling and simulating critical infrastructures and their interdependencies. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (January 2004)
2. Duflos, S., Diallo, A.A., Grand, G.L.: An overlay simulator for interdependent critical information infrastructures. In: Proceedings of the 2nd International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX 2007), June 2007, pp. 27–34 (2007)
3. Casalicchio, E., Galli, E., Tucci, S.: Federated agent-based modeling and simulation approach to study interdependencies in IT critical infrastructures. In: Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT 2007), October 2007, pp. 182–189 (2007)
4. DIESIS Project Official Web Page (2009), http://www.diesis-project.eu/ (Accessed 2009)
5. Gelenbe, E.: Users and services in intelligent networks. In: IEE Proceedings of Intelligent Transport Systems, vol. 153, pp. 213–220 (2006)
6. Gelenbe, E.: Sensible decisions based on QoS. Computational Management Science 1(1), 1–14 (2004)

7. Gelenbe, E., Lent, R., Nunez, A.: Self-aware networks and QoS. Proceedings of the IEEE 92(9), 1478–1489 (2004)
8. Gelenbe, E.: Steps towards self-aware networks. Communications of the ACM 52(7), 66–75 (2009)
9. IEEE Standard 1278, Standard for Distributed Interactive Simulation
10. Weatherly, R., Seidel, D., Weissman, J.: Aggregate level simulation protocol. In: Proceedings of the 1991 Summer Computer Simulation Conference (1991)
11. IEEE Standard 1516, Standard for Modeling and Simulation High Level Architecture
12. Calvin, J.O., Weatherly, R.: An introduction to the high level architecture (HLA) runtime infrastructure (RTI). In: Proceedings of the 14th DIS Workshop on Standards for the Interoperability of Defense Simulations, pp. 705–715 (1996)
13. Fujimoto, R.M., Weatherly, R.M.: Time management in the DoD high level architecture. In: Proceedings of the 10th Workshop on Parallel and Distributed Simulation, pp. 60–67 (1996)
14. Bononi, L., Bracuto, M., D'Angelo, G., Donatiello, L.: Scalable and efficient parallel and distributed simulation of complex, dynamic and mobile systems. In: Proceedings of the 2005 Workshop on Techniques, Methodologies and Tools for Performance Evaluation of Complex Systems (FIRB-Perf 2005), September 2005, pp. 136–145 (2005)
15. Zhao, H., Georganas, N.D.: HLA real-time extension: Research articles. Concurrency and Computation: Practice and Experience 16(15), 1503–1525 (2004)
16. Chen, D., Turner, S.J., Cai, W.: A framework for robust HLA-based distributed simulations. In: PADS 2006: Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation, pp. 183–192 (2006)
17. McLean, T., Fujimoto, R., Fitzgibbons, B.: Middleware for real-time distributed simulations. Concurrency and Computation: Practice and Experience 16(15), 1483–1501 (2004)
18. Lent, R.: Improving federation executions with migrating HLA/RTI central runtime components. In: Proceedings of the 14th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks CAMAD 2009 (June 2009)
19. Boukerche, A., Zhang, M.: Towards peer-to-peer based distributed simulations on a grid infrastructure. In: Proceedings of the 41st Annual Simulation Symposium (ANSS 2008), April 2008, pp. 212–219 (2008)
20. Riley, G.F., Ammar, M.H., Fujimoto, R.M., Park, A., Perumalla, K., Xu, D.: A federated approach to distributed network simulation. ACM Transactions on Modeling and Computer Simulation (TOMACS) 14(2), 116–148 (2004)
21. Ghanea-Hercock, R., Gelenbe, E., Jennings, N.R., Smith, O., Allsopp, D.N., Healing, A., Duman, H., Sparks, S., Karunatillake, N.C., Vytelingum, P.: Hyperion - next-generation battlespace information services. The Computer Journal 50(6), 632–645 (2007)
22. Dobson, S., Denazis, S., Fernandez, A., Gaiti, D., Gelenbe, E., Massacci, F., Nixon, P., Saffre, F., Schmidt, N., Zambonelli, F.: A survey of autonomic communications. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 1(2), 223–259 (2006)
23. Gelenbe, E., Loukas, G.: A self-aware approach to denial of service defence. Computer Networks 51(5), 1299–1314 (2007)

24. Gelenbe, E., Gellman, M., Loukas, G.: An autonomic approach to denial of service defence. In: Proceedings of the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM 2005), June 2005, pp. 537–541 (2005)
25. Oke, G., Loukas, G., Gelenbe, E.: Detecting denial of service attacks with bayesian classifiers and the random neural network. In: Proceedings of the IEEE International Fuzzy Systems Conference 2007, July 2007, pp. 1–6 (2007)
26. Boost C++ Libraries (2009), http://www.boost.org (Accessed 2009)
27. OMNeT++ Community Site (2009), http://www.omnetpp.org/ (Accessed 2009)

An Alternate Topology Generator for Joint Study of Power Grids and Communication Networks

Alpha Amadou Diallo and Claude Chaudet

Institut Telecom, Telecom ParisTech, LTCI CNRS
claude.chaudet@telecom-paristech.fr

Abstract. This article presents a method to generate conjointly topological graphs representing interdependent telecommunication and power networks. The method proposes to use a single basis, possibly random, as the common input of independent graph generation algorithms, allowing to apply different rules for different networks while preserving dependencies and interconnections' realism. It allows to generate heterogeneous networks and is evolutionary.

Keywords: Critical infrastructures protection, Topology generation.

1 Introduction

Since a few decades, operators of most large-scale infrastructures such as electrical power grids tend to rely more and more on information and communication technologies to enhance the processes of automation, control and monitoring of their infrastructures. From a convenient facility, these communication channels have become a fundamental building block, necessary to answer challenges posed by the worldwide concurrence or by population concentration in large urban centers [1]. Industrial operators have thus become dependent on efficient and reliable communication channels.

Information and communication systems therefore play a transversal and fundamental role in the management of these large systems and this relationship brings its load of unintended new vulnerabilities, as a failure or an attack on the communication infrastructure may prevent an electrical operator, for example, from controlling its own infrastructure. Moreover, several safety and security rules and assurance levels were imagined and evaluated before this tight entanglement and may require more than a minor update.

Moreover, even considering the increased monitoring and action capabilities provided by the use of telecommunication systems, large-scale failures are still possible. Concerning power grids, one may recall the electrical blackout that affected Northern America on August 14th, 2003, the Italian blackout of September 28th, 2003 or by the failure in Western Europe on November 4th, 2006. Telecommunication operators are not fully protected against these large-scale

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 50–61, 2010.

failures either, as testified by the network unavailability suffered by France Telecom on October 30th, 2004, by Bouygues Télécom on November 17th, 2004 or by AT&T in June 2007 and on September 3rd, 2008. Even though these failures may be granted to several causes, from a ship cutting an underwater cable to a software problem, they share one common characteristic. Their severity was important enough to prevent a service to be fulfilled over a long time frame.

Dependency of different infrastructures on electrical power has also slightly increased. In particular, telecommunication equipments require a power source and backup solutions such as UPS have a limited autonomy. Considering these two-directions dependencies, it is easy to imagine scenarios in which a failure in a communication infrastructure impacts the monitoring and control capacity of an electrical network, triggering protection mechanisms, which increases the number of impaired telecommunication nodes, escalating to a large blackout.

Avoiding these cascading failures requires intelligent and efficient systems, capable of detecting symptoms appearing across multiple networks while avoiding false positives. These systems should also be able to quickly take appropriate emergency responses, or at least to clearly present a status map to the operator when no immediate automatic action is possible. The conception of such an accurate general-purpose multi-infrastructures solution is therefore not immediate.

The first step in building such tools consists in identifying the vulnerable parts and setups of the network of networks composed by the multiple interacting infrastructures in order to derive representative scenarios and, only then, to evaluate response strategies. Few real scenarios exist in this domain, as situations should be avoided before they happen.

The global network we are considering is utterly complex, due to its components heterogeneity, often including software parts, to the multiple operating organization sometimes concurrent, to the number of involved equipments and to the imprecise understanding of inter-dependencies. This complexity limits theoretical modeling approaches, as the necessary choice of the simplifying hypothesis influences the results. If modeling is necessary, it shall be complemented by simulation and test-beds. Virtualization techniques, in particular, allow testing not only the algorithmic aspects of a solution, but also the software implementation, going one step further.

Regardless of the technique used to determine scenarios and to evaluate solutions, network models should be as close to real networks as possible, including their topology. However, standard topologies come in a very limited number and may not correspond to the actual evolutions of infrastructures. Most IEEE topologies for electrical networks, for instance, date from the 1960's, except for the 300-bus test case that was imagined in the 1990's. They also correspond to the United States power grid, which may not be fully representative in other countries. On the other hand, fully random topologies, which are based on some random graph models, may provide good global approximations but often lack realism regarding some aspects.

The objective of this article is to propose and describe a method to generate topologies for power grids and telecommunication networks. This technique

consists in applying network building rules, which may be different from one domain to the other, on a randomly generated map. The random basis may be used as the input of several algorithms, which allows generating related graphs and eases interconnection and dependencies identification. Section 2 presents the state of the art, section 3 introduces the algorithm we propose and section 4 presents evaluation results and a sample of the generated topologies' characteristics.

2 Related Works

Several research works have been dedicated to design topology generators, both in the electrical and telecommunication communities. Topology generation tools may be classified in two categories: they are either based on various random graphs models, or are directly infered from real network sketches.

For instance, concerning power grids, [1] use a matrix representing the North American transmission power grid to study cascading effects. In [2], authors study electromagnetic oscillations propagation in the transmission networks over two ring-based topologies and over a system of 27 buses based on the New England power system. Authors of [3] evaluate their critical points model and the cascading failures transitions over the 118 bus IEEE network and over a tree topology composed of 12 generators.

Concerning telecommunication networks, the access networks topologies are quite standard, even though there are several types of access networks, for instance xDSL or cellular networks. However, characterizing the core of the network is less easy and has been the subject of several publications. A lot of efforts have been dedicated to exploring the Internet to characterize the topology at the router level, or at the Autonomous Systems (i.e. administrative groups of routers) level. Several of these works made extensive use of the `traceroute` tool, which allows identifying the path taken by a packet from a given source to a given destination, or use BGP messages exchanges by routers to form a graph representing the Internet topology. These works lead to different network models, which are more or less realistic. In particular, the dynamic evolution of the topology is not taken into account in most of these techniques.

Recently, [4] evaluates these topologies generated using publicly available data, topologies that are widely used in the academic community. By comparing such a generated topology with real data, they show that about 10 % to 20 % of the links are omitted in the generated models for tier 1 and tier 2 Autonomous systems and that this number increases to 85 % and beyond for large networks with several peering links. This imprecision is essentially due to the short observation periods and to the number of monitors, around 400, that are supposed to discover the 26 000 autonomous systems and their links. They propose to use a combination of tools and databases to generate these graphs.

As real graphs are not easy to measure and infer, classical random graphs such as Erdös-Reiny or scale free graphs represent an appealing method to generate graphs representing arbitrary topologies. This approach is justified by a few studies published in seminal papers. Watts and Strogatz showed in [5] that

several network topologies, including power grids, exhibited a small world property, which means that they exhibit a small diameter compared to the number of vertices and that their average clustering coefficient is high. For telecommunication networks, Faloutsos *et al.* analyzed in [6] traces on the 1998 inter-AS links and showed that the subsequent graph's degree distribution fitted a power law. They showed, in particular, that only 40 % to 50 % of BGP nodes belonged to tree-like topologies and that among these trees, 80 % exhibited a depth of only 1 level, never exceeding 3 levels. Most vertices belonged to fully meshed graphs. Therefore, few long links existed.

Models based on a specific class of random graphs called *random geometric graphs* have gained much popularity in telecommunication networks modeling. Waxman proposed in 1988 [7], to base topology generation tools on the Euclidian distance between nodes. Concerning power grids, Wang *et al.* propose in [8] to generate electrical network whose nodes locations are distributed according to any probability distribution, using a distance interval, rather than a bound, to determine whether two nodes are connected or not. In [9], Krause generates of graphs derived from a reference by replacing certain links originating at random nodes by links between this node and its two-hops neighbors, preserving the characteristics of the graph such as degree distribution or clustering coefficients.

However, some studies on the influence of the hierarchical structure of the Internet on the topological graph, such as [10] showed that the Waxman model was not entirely realistic. Recently, an article [11] went one step further, proposing to extend a reference graph to generate a random topology, preserving its main characteristics such as the degree distribution [12].

Most of these random graphs models can be further parameterized to obtain graphs with the desired properties, such as the distribution of degrees, betweenness centrality or clustering coefficients. However, random graph models are often criticized, as they are built in an engineering-agnostic way, which limits their realism. For example, if the degrees distribution of real networks fits to a power law, saying that any power law-based graph may represent a realistic network topology is false, especially when the maximum degree of a particular vertice exceeds the limitations imposed by hardware.

3 Topology Generation Technique

3.1 Justification and Requirements

The contributions mentioned above propose several topology generation algorithms, to which one may address more or less criticism regarding their accuracy. However, independently of the pertinence of the different methods, none of them seems fully suited to the conjoint study of telecommunications and power networks, as they do not provide a way to specify, identify and focus on interconnections and dependency relationships between networks. We identified three main points that we expect from such a multi-topologies generations tool:

Ease interdependencies identification and specification. Examining the interactions between an electrical network and a telecommunication network

requires using two different graph generators. As the constraints influencing the design of these networks are slightly different, using the same generation tool for both networks would probably lead to inaccuracy in the general case. If there is no particular difficulty in generating two separate graphs, deciding on their interdependency is difficult unless both graphs derive from a common basis, as dependencies may come from logical constraints as well as from other factors such as the proximity of two particular network elements.

Ease interconnections identification and specification. Administrative information is generally absent from the generated topologies. However, the Internet and the global power grids are both composed of several small networks, interconnected through particular nodes and links, for instance BGP nodes in IP networks. Each of these individual networks has its particular characteristics and its own management policy, which may have a strong impact on failures propagation. A realistic topology generation tool shall identify individual networks boundaries and shall be coherent with common practices regarding individual networks shapes and interconnection points placement. The topology generator shall also be able to mix and interconnect small and large networks to reflect operators' heterogeneity.

Preserve identification of elements to enhance post-simulation analysis. An element dedicated to the interconnection of multiple operators shall be clearly identified, as its role, importance and behavior are different from the role of an element internal to an operator's network, and this information should not disappear once the global graph is produced. Providing such identification may be a requirement of certain algorithms using the generated topologies or not, but it certainly allows applying a coherent policy regarding graph parameters affectation, for instance links capacities. Without such labeling, links capacities are either drawn randomly, or selected according to local graph characteristics such as vertices degrees or betweenness, which may reflect the real conditions or not. Elements shall at least be grouped in classes and given labels, in order to allow defining levels of importance.

If the latest references mentioned in section 2 allow generating topologies closer to real networks, they do not fully meet the aforementioned constraints. [8], for power grids, does not allow distinguishing and annotating elements, abstracting the operator in charge of a node. Moreover, domain practices such as the N-1 rule enforcement have to be tested after graph generation, as they do not influence the generation process. [11], addressing telecommunication networks, is not fully satisfying either when it comes to interconnection of different networks, as localization information, or equivalent, is missing.

The following sections describe the strategy we chose to generate communication networks and electrical networks, individually and outline a method to interconnect infrastructures at a national level. In brief, we try to take the opposite course to the aforementioned random techniques that generate the graph and, once this first step is realized, place particular nodes such as servers and BGP routers in a telecommunication network, or generators and loads in a power

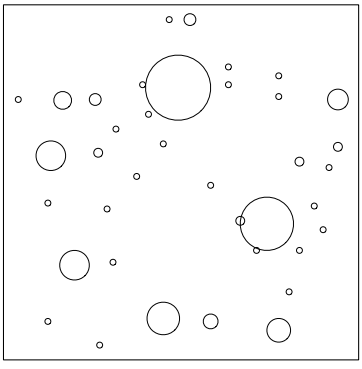


Fig. 1. Example geographical map – circles sizes represent the sites' weights

grid. Even though the result may be identical, it seems logical to reverse this process, placing the particular nodes first and, only then, defining their interconnection. From this observation and in order to provide a meaningful and logical interconnection process, we derive both electrical and telecommunication graphs generation algorithms from a common basis, which is a random country map. Our country model simply consists of cities localizations over an area with random population counts, on which we deploy and interconnect both networks.

The first step therefore consists in generating the reference country map, which simply consists in an area in which cities are randomly positioned and populated. However, any distribution is possible. We therefore obtain a set of N sites with associate geographic coordinates $(x_i; y_i)$ and weights (w_i). The different weights may reflect the population counts of the different cities, the activity in different area, the business opportunities represented by different regions or any other combination of high-level criterion that influences organizations that decide to build or extend a network infrastructure. Figure 1 represents such a map, the size of the bullets reflecting the weights of the different sites. For the results presented in this paper, we used uniform distributions for the x and y coordinates, as well as for the weights of the different nodes.

3.2 Communication Networks

Communication networks are dimensioned according to human activity. Interconnection is denser in large populated areas, close to business activity regions and nearby interconnection points such as BGP routers. Today, operators build their networks building and evolutions on market studies, starting from areas with a high potential and extending the network either in its coverage, or its capacity. There are no dedicated traffic sources nodes, as popular websites, for instance are generally replicated over several locations for load balancing using, for example content delivery networks.

The generation algorithm takes as input the weighted geographic map described above. It first selects the site with highest weight node, which often corresponds to the largest urban center. This site will have a high degree in the generated network topology, as it regroups a large number of users. This site is then connected to secondary sites, i.e. sites with a smaller weight.

To select such secondary sites, several strategies may be imagined. One may define a weight threshold above which a site is considered as secondary, or one may define that the number of secondary cities is either fixed, or a proportion of the total number of cities. We arbitrarily chose to divide the geographic area in 9 areas and to elect at most one regional capital per area (Fig. 2(a)). If no city in an area possesses a weight higher than a fixed threshold, the region does not contain a secondary site. Finally, secondary sites that share a common coordinate (i.e. same column or same line in the 3x3 division of the map) are connected together, preferring closer interconnections (Fig. 2(b)). For every region, the algorithm examines if the immediate next region has a secondary site, turning clockwise. If such site exists, then we interconnect the two close sites (Fig. 2(c)).

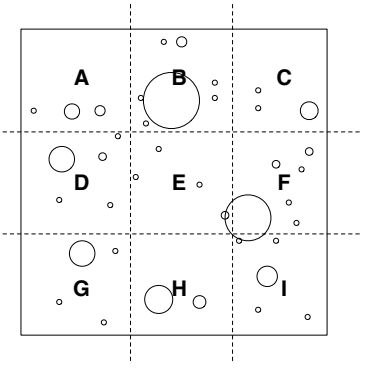
Communication lines between the capital and regional capitals pass through every intermediate site close enough to the direct path between these two sites (Fig. 2(d)). The decision to deviate a link to connect an intermediate node is based on the ratio of the distance of the intermediate node to the link over the total distance of the straight line link. Once this step is finished, remaining sites are interconnected to the network by a direct communication line towards the closest connected node (Fig. 2(e)). Additional nodes may be added along each link to enforce a maximum link distance, modeling the presence of physical constraints on the links lengths if required.

Once all nodes have joined the network, the algorithm searches the graph for long bridges. The bridges whose removal disconnect more than one single node are unlikely to appear in a real topology, as they represent single points of failures and are contrary to most common practices such as the N-1 rule that influence the design of networks. To suppress these bridges, we connect their end nodes either to the core network, or to another bridge, depending on the closest node, which ensures that the augmented graph does not contain any more long bridges (Fig. 2(f)). We do not connect single nodes that are connected to the network by a single link, though, as these situations may appear.

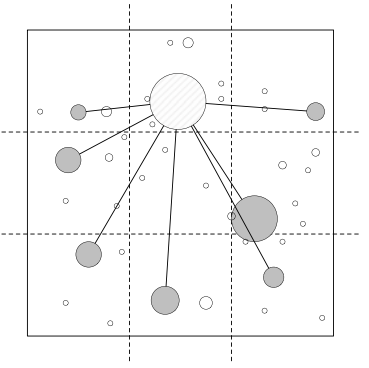
Figure 3 represents an example resulting topology, along side with a real French topology. To ease the correspondence cities coordinates and populations have been given as an input to the algorithm and correspond to the real geographical data of this country.

3.3 Electrical Networks

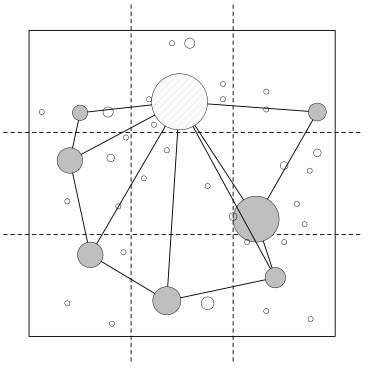
Electrical networks are generally composed of three classes of elements. Production elements are spread over the territory, generally relatively far away from houses to mitigate transportation cost and hazards they represent. Loads, or consuming nodes are directly function of the activity and are somehow close to the population distribution, except from great industrial complexes. The rest



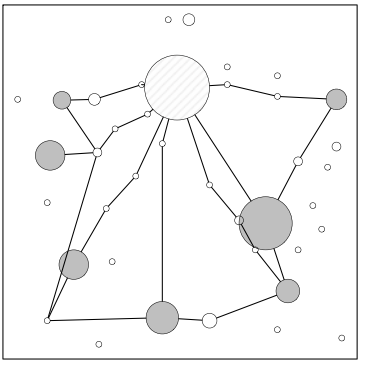
(a) Step 1 : divide the map into 9 equivalent areas



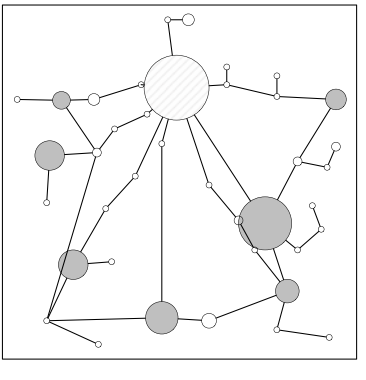
(b) Step 2 : plan to connect the capital site with secondary sites



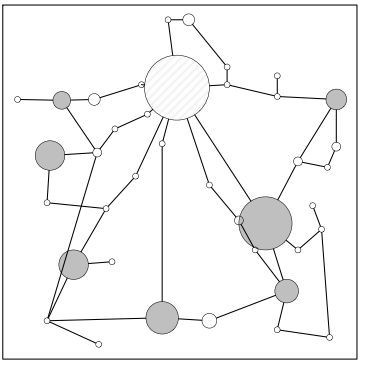
(c) Step 3 : interconnect close regional capitals together



(d) Step 4 : Create real links by deviating long lines through close sites

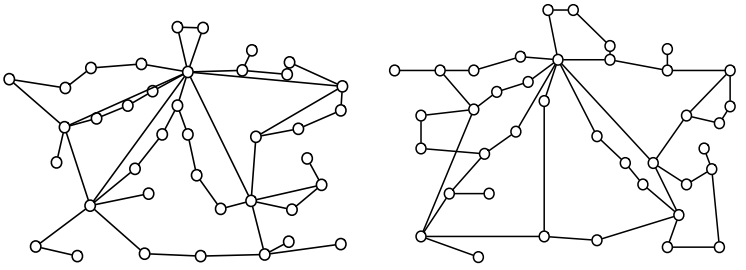


(e) Step 5 : Connect remaining sites



(f) Step 6 : Interconnect pending branches (bridges)

Fig. 2. Generation process of a telecommunication graph



(a) Real Network Graph (b) Generated Network Graph

Fig. 3. Telecommunication Networks: Comparison between the generated and real topologies

of the infrastructure is composed of transmission and transformation elements. In this work, we only generate the part of the transmission network that corresponds to the heart of the network. The distribution networks have a topology slightly different from the transport network, as they are organized as trees, bearing almost no redundancy and as their failure generally concerns a limited sets of users.

The generation of electrical grid networks follows the same principles as the telecommunication networks, based on the same coordinates and weights. The principal difference between theses two algorithms lies in the order of interconnection of the different nodes. Nodes that represent production centers are connected first regardless of the zone they belong to.

Then, other sites, representing transformers and loads, are interconnected in turn, regardless of the weights of the different sites of the map, as the algorithm respects the N-1 rule at the topological level due to the long bridges suppression. We chose not to take into account the weights in our generation process, as we treat every load equally. However, they could be used to specify that some sites consume or produce more than others if required.

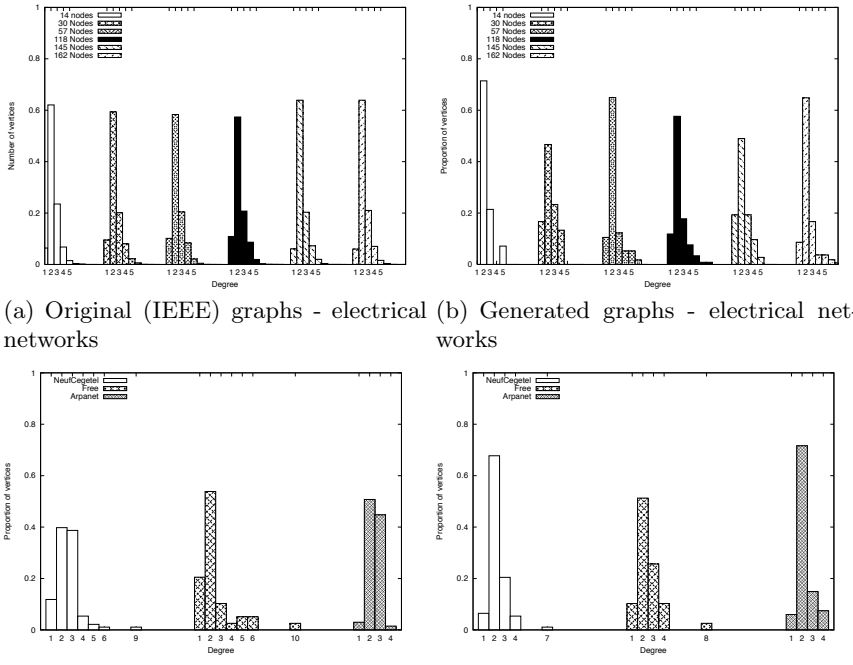
3.4 Interconnections

From the precedent descriptions, interconnections between different networks are straightforward. Interconnections between networks belonging to the same domain are located close to high weights elements and to the map boundaries. Dependencies and interdependencies between infrastructures of different domains can be derived from the geographical coordinates.

4 Algorithm Evaluation

4.1 Degree Distribution Comparison

The degree distribution of the different vertices reflects the distribution of the importance of the different nodes. For electrical networks, Figure 4(a) represents the



(a) Original (IEEE) graphs - electrical networks

(b) Generated graphs - electrical networks

(c) Original graphs - telecommunications

(d) Generated graphs - telecommunications

Fig. 4. Comparison between generated and real graphs degree distributions

degrees distributions of different classical IEEE bus networks. The x-axis represents the degree of the different vertices while the y axis the proportion of vertices bearing this degree. The different pack of bars represent different scenarios. Figure 4(b) presents the degree distribution of the corresponding generated graphs. If the proportion of vertices may vary from one graph to the other, which is essentially due to the limited original data-set, we can observe on these figures that the basic properties of this distribution are preserved. The peak of the distribution is situated at a degree of two and distribution of the core nodes' degree (i.e. not considering the leaves) fits a power law. Figures 4(c) and 4(d) compare the degree distributions of real telecommunication networks with the ones generated by our tool.

A few classical metrics may also be useful to verify certain properties on the generated graphs. First, the width of the graph should increase with the number of nodes, obviously, but this growth should not be too fast, as it would indicate that the algorithm tends to generate long paths. It may indeed preserve the degrees distribution which has a peak around the value of 2, but would not be quite realistic, as networks are generally meshed due to redundancy, thus reducing their width. We examined both the network diameter and the closeness of the different vertices (i.e. their average distance, in number of edges,

from other vertices). Both parameters tend to grow linearly with a rather small factor (about 0.1 for the diameter and 0.03 for the closeness).

Finally, as redundancy is a key factor to both types of networks design, we verified that the algorithm preserved this characteristic. We chose to express structural redundancy through the number of fundamental cycles present in the graph, which is reflected by the α index, defined as the ratio between the number of such fundamental cycles ($|E| - |V| + 1$ in a connected graph) and the number of possible fundamental cycles ($2 \cdot |V| - 5$). For all generated graphs, this ratio is about 0.1, regardless of the network size, which means that the generated graphs have about 10 % of the potential number of cycles they may have. This metric does not evolve with the number of vertices, indicating that large graphs have the same structural properties as smaller ones, which corresponds quite well to classical rules influencing the design of networks (for instance the N-1 rule).

5 Conclusion

In this article we depicted and evaluated a method to generate graphs representing the topology of electrical and telecommunication networks. This algorithm bases the graph generation on shared data, namely a random population map, easing interconnections between several networks. Another advantage of this mechanism lies in the possibility to use different stochastic processes to generate the population maps, better reflecting the networks deployment processes in different countries, but also different edges creation policies, allowing to reflect, for instance, the United States case, whose power grid is less centered on a few particular cities. The de-correlation between the two aspects makes the algorithm flexible and evolvable.

The first characterizations indicate that the generated graphs have the intended properties, even though further tuning is necessary to be able to generate graphs structurally identical to the real world topologies. Further developments of this work include reflections around the nodes-zones association process which is currently based on raw distances, and the full specification of the interconnections process between networks of the same type to reflect structures such as the BGP routers present in the Internet.

Acknowledgments

This work has been partially funded by the European Union project IRRIIS (Integrated Risk Reduction of Information-based Infrastructure Systems) under the IST programme of the Sixth Framework Programme (FP6-2005-IST-4).

References

1. Kinney, R., Crucitti, P., Albert, R., Latora, V.: Modeling cascading failures in the north american power grid. The European Physical Journal B - Condensed Matter and Complex Systems 46(1), 101–107 (2005)

2. Parashar, M., Thorp, J.S., Seyler, C.E.: Continuum modeling of electromechanical dynamics in large-scale power systems. IEEE Transactions on Circuits and Systems 51(9), 1848–1858 (2004)
3. Carreras, B.A., Lynch, V.E., Dobson, I., Newman, D.E.: Critical points and transitions in an electric power transmission model for cascading failure blackouts. Chaos 12(4), 985–994 (2002)
4. Oliveira, R.V., Pei, D., Willinger, W., Zhang, B., Zhang, L.: In Search of the Elusive Ground Truth: The Internet's AS-level Connectivity Structure. ACM SIGMETRICS Performance Evaluation Review 36(1), 217–228 (2008)
5. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world' networks Nature 393 (1998)
6. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication (Sigcomm), Cambridge, United States, August 1999, pp. 251–262 (1999)
7. Waxman, B.M.: Routing of multipoint connections. IEEE Journal on Selected Areas in Communications 6(9), 1617–1622 (1988)
8. Wang, Z., Thomas, R.J., Scaglione, A.: Generating random topology power grids. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences, Waikoloa, Hawaii (January 2008)
9. Krause, A.: Generating networks with realistic properties: The topology of locally evolving random graphs. In: Proceedings of the 1st International Conference on Economic Sciences with Heterogeneous Interacting Agents (WEHIA 2006), Bologna, Italy (June 2006)
10. Zegura, E.W., Calvert, K.L., Bhattacharjee, S.: How to model an internetwork. In: Proceedings of IEEE INFOCOM 1996, the Fifteenth Annual Joint Conference of the IEEE Computer Societies, Networking the Next Generation, San Francisco, USA (March 1996)
11. Mahadevan, P., Hubble, C., Krioukov, D., Huffaker, B., Vahdat, A.: Orbis: Rescaling Degree Correlations to Generate Annotated Internet Topologies. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM), Kyoto, Japan (August 2007)
12. Mahadevan, P., Krioukov, D., Fall, K., Vahdat, A.: Systematic topology analysis and generation using degree correlations. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM), Pisa, Italy (September 2006)

Trouble Brewing: Using Observations of Invariant Behavior to Detect Malicious Agency in Distributed Control Systems

Thomas Richard McEvoy[1] and Stephen D. Wolthusen[1,2]

[1] Information Security Group, Department of Mathematics, Royal Holloway, University of London, Egham Hill, Egham TW20 0EX, UK
[2] Norwegian Information Security Laboratory, Gjøvik University College, P.O. Box 191, N-2802 Gjøvik, Norway

Abstract. Recent research on intrusion detection in supervisory data acquisition and control (SCADA) and DCS systems has focused on anomaly detection at protocol level based on the well-defined nature of traffic on such networks. Here, we consider attacks which compromise sensors or actuators (including physical manipulation), where intrusion may not be readily apparent as data and computational states can be controlled to give an appearance of normality, and sensor and control systems have limited accuracy. To counter these, we propose to consider indirect relations between sensor readings to detect such attacks through concurrent observations as determined by control laws and constraints.

We use a brewery bulk and fill pasteurizer as a specimen for biochemical processes. We motivate our approach by considering possible attacks and means of detection. Here we rely on the existence of non-linear relationships which allow us to attach a greater significance to small differences in sensor readings than would otherwise be the case and demonstrate the insufficiency of existing sensor placement and measurement frequency to detect such attacks.

Keywords: SCADA, DCS, anomaly detection, pasteurizer, non-linear relationships.

1 Introduction

Sensors and actuators in modern distributed control systems (DCS) are increasingly general-purpose computing platforms, replacing simple purpose-built components with Intelligent Electronic Devices (IED), both hardware and software, which are COTS. These systems, which are directly connected to TCP/IP-based networks, must have a full network protocol stack and operating system [1]. This alteration makes these systems increasingly vulnerable to external attack as well as internal sabotage.

Recent research on the security of supervisory data acquisition and control (SCADA) and DCS systems has focused on intrusion, particularly anomaly detection at the protocol level based on the fact that traffic on networks

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 62–72, 2010.

interconnecting SCADA/DCS components should be well-characterized and hence particularly amenable to anomaly detection techniques – for example, [2].

We argue that during attacks in which the sensors and actuators themselves have become (logically) compromised or physically manipulated, anomalies would not be so readily apparent [3]. We have also demonstrated previously that it is possible to conceal such attacks in noise which is inherent in measurement processes and not amenable to elementary statistical analysis [4].

We propose an approach, whereby, by defining invariant behavior in the control plant in terms of indirect relations between sensor readings, we may, using sensors already present and, where appropriate, additional software or hardware based sensors – determine if there has been a violation of these invariants.

In particular, we may make use of non-linear relations between sensor readings where deliberate attempts to conceal differences of actual from predicted readings on a limited subset of potentially compromised sensors will create obvious anomalies when considering the larger set of readings.

We assume for the purposes of this study that the attacker has suborned a subset of sensors and is seeking to conceal his manipulation of the system behind falsified readings or using process "noise". Such manipulation may occur over a prolonged period in order to create long term damage or prolonged exposure to economic risk on the part of the victim.

We discuss the type of subversion attack using the example of a bulk and fill pasteurizer found both in the food and pharmaceutical industry whose failure or improper operation would lead to severe product degradation and contamination. A characteristic of this type of control system is that even small deviations in actual parameters can result in severe consequences without being readily visible to operators.

Our findings apply more widely to any systems where subversion may be used as a preparatory step for a complex attack. The proposed mechanism would allow the early identification of invariant violations and anomalies from normal operation expected under the control law in effect, providing the means to discover physical or cyber subversion attempts in early stages.

The remainder of this paper is organised as follows: Section 2 discusses related work in this area. Section 3 describes the problem of sensor and actuator subversion. Section 4 describes the proposed detection mechanism. Section 5 provides a descriptive analysis of the pasteurizer and its control processes. Section 6 gives a potential attack model. Section 7 details the detective approach, and we conclude with a discussion of our results, brief conclusions and an outlook on future work in section 8.

2 Related Work

SCADA and DCS systems have become increasingly vulnerable to cyber attack due to modernisation and exposure to untrusted networks [5,6,7,8]. Moreover, economic pressures have also increased the interdependence of these networks

and their exposure to both internal and external threats which underlines the seriousness of this threat [9].

Approaches to intrusion detection divide between signature-based approaches, generally at perimeter defences, and anomaly-based which address insider threat and direct attacks against RTUs, sensors and actuators [9,10].

The predictable nature of SCADA protocol and usage is perceived to be an advantage in detecting system anomalies [3,10,11], but the limitations of this approach in the face of knowledgeable attackers capable of manipulating computational states or utilising signal noise to obfuscate attacks have been clearly recognised [3,4].

We argue that this highlights a requirement for the use of multiple sensors [2,4,12] which make use of differing points of view [3] for anomaly detection. This requirement is underlined by the introduction of sophisticated control processes which rely on multivariate controls and hence require more complex forms of supervision [13] although the approach would also apply to more traditional control systems.

Our method has some parallels in the Safeguard project [9] making use of multiple agents, but the latter approach is weakened by a failure to account for time differences in distributed computing systems, where a knowledge of causal and temporal may be critical to understanding whether apparently conjoint events are genuinely critical in nature or if apparently isolated events implicated in serious faults. So our work relates strongly to fault detection in SCADA and DCS environments [14].

We seek to combine a knowledge of causal effects, which may be modeled in this case using non-linear structural equations [15], with a lightweight approach to observation which is capable of viewing data about physical processes and computational states in parallel [16]. In this paper, we concentrate on establishing and motivating our approach to observing and comparing sensor readings and determining what issues arise in modeling such systems and in practical implementation.

3 Problem Description

Intrusion detection efforts have concentrated on detecting anomalies in SCADA protocol usage which is more predictable than that used in general networks [2]. However, two issues arise. First, an attacker may use the complexity of the operating systems or networks to falsify data communication or command statements [3]. Second, assuming the actuators themselves are, at least, partially suborned, he may hide an attack in the process "noise" [4].

This opens up the possibility of the attacker manipulating the system to give the appearance of normality using a subset of sensors or signal obfuscation and either launching an attack by directly subverting the operation of the system or else misdirecting the efforts of human operators or other connected systems based on the manipulated presentation of the sensor data. Clearly, the actual attack model will vary dependent on the nature of the process - see section 6 for our specimen process.

4 Approach

We assume therefore that during an attack, we will be dealing with a control system where at a given point, some, but not all, sensor readings may be subverted or system state obfuscated by the subversion of a limited set of controllers. We, therefore, need to show that we can make aggressive use of indirect relationships which exist, but are not normally considered for control purposes, between sensor and readings to uncover anomalous results. Ideally, our approach must preserve real-time performance characteristics by utilising a conservative approach to taking observational readings.

Based on previous work in the general case of distributed systems [16], we propose the use of additional lightweight observer processes which may be associated with sensors and/or actuators, or independent of them as required, both in hardware and software.

While the basis for detection are the control laws used, we seek to derive relations which indirectly evidence the correct (or otherwise) operation of the control law. The existence of these relations, which we focus on in this paper, drives any augmentation of the observation process.

It follows that, during an attack, assuming a partition of sensors is suborned, that the observation mechanism should provide contrasting pictures of what is occurring in the system and this contrast provides the basis for anomaly detection.

We describe the kind of attacks which may occur and the issues arising for modeling and implementation using a bulk and fill pasteurizer used in a brewery. This stands as a type for biochemical control systems, but some of the findings will be transferable to other systems, particularly in the energy production industry, where complex heat exchange requirements exist.

5 Operation of a Pasteurizer

5.1 Operation of a Pasteuriser

A bulk and fill pasteuriser divides into four sections. An initial section (*splitter*)where the already pasteurized product heats the product entering the tank. The pre-heated product passes through a second heat exchange chamber, where heated water, under pressure, is passed through pipes, to bring it to its target temperature. This water is itself heated in a third chamber of the pasteurizer using controlled bursts of steam to maintain its temperature. Once heated, the product[1] passes to holding pipes to maintain its temperature, completing the pasteurization process, and circulates to a *regeneration* chamber, co-located with the splitter, where it exchanges heat with the product entering the pasteurizer. Finally, it enters a cooling chamber, where glycol refrigerant is piped under pressure to cool to kegging temperature. Before the start of any kegging run, the pasteurizer will be sterilized to avoid microbrial contamination.

[1] In our example, this is an Irish stout beer, reflecting national differences in critical infrastructure categorization.

The goal of pasteurisation is to kill off a sufficient number (98%) of microbial contaminants to guarantee product shelf life for a specified period of time under uncontrolled circumstances[2]. Failure to achieve the correct "kill rate" measured in *PU*s (*pasteurizing units*) may lead to early product contamination, product recall, and subsequent severe economic and reputational damage to the company.

The operation is illustrated in figure 1:

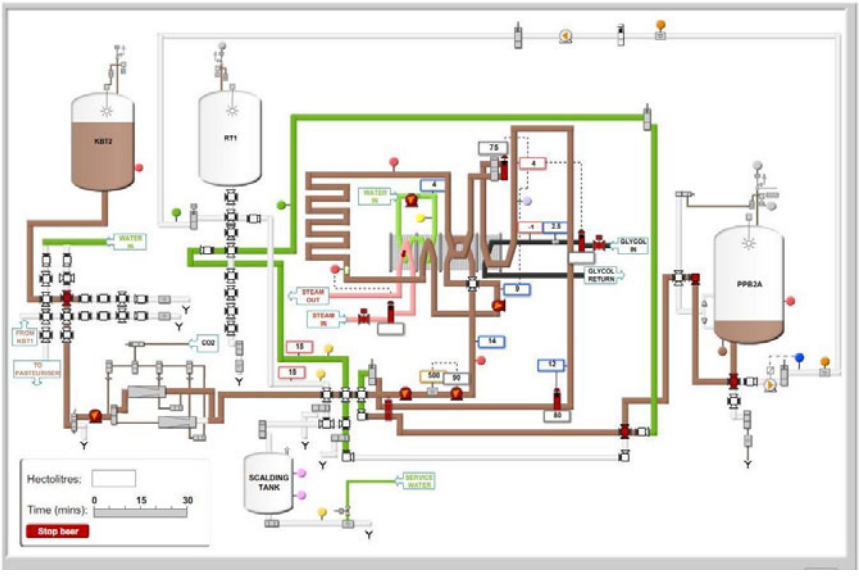


Fig. 1. Flash Pasteurizer

5.2 Pasteurisation Requirements

The primary control concern is to maintain the correct pasteurizing unit (PU) value. The relationship between PU flow rate and heating temperature is captured by a non-linear equation –

$$PU = HR * 1.393^{(T-60)} \tag{1}$$

where PU = Pasteurisation unit, HR= Heat retention time (min) and T = Heating Temperature. The PU value is clearly subject to even minor variations in temperature and, to a lesser degree, variations in the flow rate, a fact which can be exploited by an attacker (see Section 6). The heat retention time is equal to the ratio of heat retention volume V to the total flow rate Q and $HR = (V/Q)*60$ [3]where HR= Heat retention (min), V = Heat retention volume (Hl), Q = Total flow rate (Hl/Hr).

[2] i.e. under arbitrary control and transport conditions.

[3] V/Q results in an inverse time unit and is a space time constant which may be used in analysing dynamic response.

The relationship between the flow rate Q and the PPBT (Post Pasteurisation Buffer Tank) level is described as follows. Let Q be the Flow Rate. Let L be the buffer tank level then –

$$Q = Q_{max} - Q_{min} \times \frac{(L_{actual} - L_{min})}{L_{max} - L_{min}} \tag{2}$$

The flow rate is clamped to maximum and minimum values, where the product level (measured approximately by weight) crosses its maximum and minimum values respectively – for example, $Q_{max} = 500$, $Q_{min} = 250$, $L_{max} = 220$ and $L_{min} = 100$. As the beer kegs are filled concurrently with pasteurisation, the aim is to balance the output to the kegs and input flow rates to create a steady product flow throughout the pasteurizer unit and minimize requirements to alter temperature.

The aim of pasteurisation is to approach the nominal PU value. In the event of failure to do so, the divert PU value and abort PU value are product dependent as is the nominal PU value.

The temperature control hotside preset is derived from the actual flow rate FR and the PU setpoint according to equation 1. Re-arranging the formula in terms of T gives –

$$T = \frac{\log PU - \log(\frac{V \times 60}{FR})}{\log 1.393} + 60 \tag{3}$$

It should be noted that the process is sufficiently sensitive that a drop of more than 1^o C in temperature would be considered to breach product specifications. A breach in product specification over time increases the risk of a product recall, which would be economically and reputationally disasterous.

The cooling controller acts independent of flow rate. Its function is to maintain a constant product outlet temperature of between $7 - 10^o$ C, regardless of flow rate. Two sensors are used, one to measure the temperature of the glycol mix used as a refrigerant, the other to measure the cold side temperature of the beer. Glycol is released as required to adjust this temperature.

5.3 Heat Exchange Characteristics

The main activity in the pasteurizer is heat exchange. This is so efficient that minimal effort is required to heat the product on the hotside to gain optimal temperature and similarly to cool it on the cold side. For practical purposes, this can be modeled as a relative gain or loss, depending on whether we are considering cold or hot fluids respectively. Hence,assuming a steady flow rate and not knowing the area of the heat exchanger, we might approximate heat exchange using the equations $T_{cold} = T_{cold} + (T_{hot} - T_{cold})(E)$ where T is temperature and $T_{hot} = T_{hot} - (T_{hot} - T_{cold})(E)$ where E is the efficiency ratio for the heat transfer. This model of heat exchange is only approximate and more refined models exist – a fact which will be key to subsequent analysis (see also section 8).

6 Attack Model

Assuming the utilisation of subverted sensor data, two types of attack vectors can be identified in this specific configuration. The first is to create a situation where inefficiencies are introduced in the operation of the pasteurizer with the aim of creating periods where extended maintenance efforts are required to support operations with subsequent loss of production time. The second is to place the quality of the product at risk. Over a prolonged period there would a subsequent increase in the expectation of product recall leading to the possibility of subsequent reputational damage to the company.

Attacks on the efficiency of the pasteurizer could include aspects such as altering pressure differentials in water and glycol pipes to lower the effectiveness of heat transfer or deactivating one of the product pumps to create greater fluctuations in the flow rate [17]. Dependent on the sensitivity of the product[4], overheating the product could lead to fouling and loss of efficiency in the heat exchange. A more direct attack might involve creating the circumstances during sterilisation where the temperature differential between hot and cold side has the potential to cause thermal shock with subsequent cracking and spillage in the heat exchange plates. Combining efforts of this nature can result in prolonged efforts at fault analysis and maintenance with subsequent loss of production to the plant. Moreover, by simulating failures, operators may be induced to discard legitimate product that is assumed not to meet specifications.

Some attacks on efficiency also have the potential to degrade product quality. For example, an overheated product (here e.g. a light lager), dairy product or condiment will lose its texture rapidly before or during storage and may not have desirable taste. A more robust product such as the one selected for our example would not be so prone to this kind of attack, but a failure to achieve sufficient PU opens it to the risk of loss of shelf life, since storage and transport may not be carried out under ideal conditions[5]. Finally, any detection of a failure to meet product specifications even if not immediately consequential will lead to recall of the product batch with subsequent knock on effects for the company's profit and reputation – in addition to any legal jeopardy. In each case, the attacker may make use of false sensor data, combined with direct or indirect actuator manipulation to achieve his results. For example, significantly overreporting the flow rate of the pasteurizer while manipulating its value downwards using buffer tank information will lead to lower temperatures being used than are required for pasteurization. Similarly, one could directly conceal a manipulated temperature loss. Depending on product sensitivity (products with lower PU are more sensitive to temperature change) it may be sufficient to obfuscate sensor readings using process "noise" to conceal the attack.

[4] This is not applicable for the product described here, but is relevant for products such as pharmaceuticals also relying on pasteurization.

[5] We note that this is particularly effective for products where loss of quality and shelf life is not immediately obvious, e.g. in pharmaceutical products.

7 Detection Model

For detection purposes, we define the relationships between sensor readings using non-linear structural equations to build a causal model [15][6]. We refine this model by replacing the equations with well-defined statistical functions, incorporating probabilistic disturbances and time lags, which represent invariant relations. Knowledge of these relations enables us to analyse possible indirect effects of an attack. Essentially, we build a "memory" into the model. This approach may be regarded as a mirror image of MPC, but with the purpose of seeing where we have been and hence, assuming normal operation of control laws, where we should be now [18].

In the following example we use several abbreviations in the interest of convenience, namely INIT for "initial", PX, for "post heat-exchange", HS for "hot-side" (where subscripts indicate initial(1) or post holding pipe(2) values), and CS for "cold-side"

Let S, T, U stand for temporally succeeding product temperature states. Let W be water temperature. Let P be any preset temperature. Let FR be the flow rate. Let G be the glycol temperature. Let ϵ stand for any "disturbance", representing any empowering, or inhibiting factor which affects probable state. Let $[ACTUATOR]$ be the name of any actuator implicated in the system. We use actuator labels as a gloss to the model to indicate where we might predict their activity. Let $f, g, \ldots$ be any structural causal function. Let $=$ indicate the relation "results causally from"

We hence obtain

$$T_{INIT} = f(SEASON, \epsilon_1) \tag{4}$$
$$T_{PX_1} = f(T_{INIT}, S_{HS_2}, \epsilon_2) \tag{5}$$
$$T_{HS_1} = f(W, T_{PX_1}, \epsilon_3)[STEAM] \tag{6}$$
$$PU = f(FR, T_{HS_1})[PU] \tag{7}$$
$$T_{HS_2} = f(T_{HS_1}, \epsilon_4) \tag{8}$$
$$T_{PX_2} = f(U_{INIT}, T_{HS_2}, \epsilon_5) \tag{9}$$
$$T_{CS} = f(T_{PX_2}, G, \epsilon_6)[GLYCOL] \tag{10}$$

We can therefore define an attack abstractly as an "intervention" in the causal model, which may be achieved by falsifying states or functionality and concealing these changes from the operator or other parts of the system. We argue that such an attack may be detected by utilising knowledge of how these indirect relations act even where a subset of sensors has been compromised.

This approach is based on the reasonable assumption that biochemical processes contain non-linear relationships. These relationships have the consequence that small deviations in value have a greater significance that they would for purely linear relationships. For example, the relationship between flow rate, PU

[6] Not to be confused with non-linear relations which enable other aspects of our approach.

and temperature is one such relationship. So if we are able to use an indirect relationship such as the cold side temperature to estimate changes in the hot side temperature (via the flow rate) we can show where a hotside temperature violation has occurred.

To be precise, each subequation represents in the Laplace (or Fourier - depending on the needs of analysis) domain, a transform $G(s)$ and a disturbance (which may be modeled as random or colored noise). Pursuing our example, a step change $Au(t)$ to the hotside temperature T_{HS_2} should therefore result in a dynamic response in the coldside temperature T_{CS} and subsequently affect the operation of the glycol valve. The presence of a response in the absence of its cause becomes a reason for suspicion. However, this suspicion must necessarily be expressed in probabilistic terms, depending on the significance of the response. Intuitively, where the response is governed by a non linear relation the "sensitivity" of the process to change becomes easier to "read". This does not preclude dealing with linear relations with sufficient gain in their dynamic response.

Such an analysis however requires a more sophisticated model of how the pasteurizer works. That is, the approximations to behavior, in (for this example) the heat exchange systems which are sufficient for production purposes are not sufficient for intrusion detection purposes. The basic formulae for heat exchange – see section 5.3 – assume a steady flow rate (Even a more complex version of this formula, based on a knowledge of the area of the heat exchange, allowing the calculation of log mean temperature differences makes the same assumption.) A more sophisticated version of this relationship is required which takes into account the dynamic behavior of the flow rate, as a minimum, dealing with partial differential equations based on behavior over time - for example [19,20] - and this work may need to be further refined by simulation to account for disturbances.

It may also be informative to have additional sensors – for example, at interim stages in the heat exchange process where a loss of heat energy equates to a loss of information[7] and on the return pipe for the glycol flow where an increase in temperature may also provide evidence for detecting obfuscated alterations to process parameters.

Setting aside issues of compliance and licensing (although these may be significant), these considerations do not present the computational barrier which they might have constituted a decade ago. The introduction of multiprocessor systems as a *de facto* standard means there is scope to utilise additional processes distinct from the control process to take supplementary readings at a more frequent rate without necessarily affecting performance. It is also cheaper to deploy additional hardware sensors and associated IEDS. Nonetheless, given that we wish to incorporate such systems into existing units as well as new plant, we have to be careful to minimize the efforts involved and have to achieve a compromise between detection capabilities and the practicalities of additonal sensor deployment. However, we believe the additional effort to be justified by the threat.

[7] In contrast to the usual order in computer systems where loss of information leads to heat energy through dissipation.

8 Discussion and Future Work

In this paper, we have discussed the potential for knowledgeable attackers to conceal the presence of their attacks by manipulating computational states, actuator settings and data values on SCADA and DCS systems. In the light of this potential, we have proposed the utilisation of a novel observation mechanism which acts independently of process control requirements which garners data about the current state of the system through comparing observations with a refined model of the system which consider indirect invariant relations between sensor values.

Both the determination of sufficiently refined inter data relations and the need to balance detective ability with performance, cost and compliance requirements mean that extensive efforts are required in simulating the model and in considering implementation choices. In part these requirements are set by the conservative nature of control systems which require compromise on detection sensitivity and model complexity as well as employing the minimum necessary additional sensors or readings. Extensive modeling is required for design and implementation. However, the increasing threat to CI systems justifies these efforts.

Future work will concentrate on experimental work both in simulation and on test rigs to verify the validity of our approach both on the current example and on other biochemical systems and to address the issues we have raised.

References

1. Creery, A., Byrnes, E.J.: Industrial Cybersecurity for Power System and SCADA Networks. In: Proceedings of the 52nd Annual Petroleum and Chemical Industry Conference, Denver, CO, USA, pp. 303–309. IEEE Press, Los Alamitos (2005)
2. Coutinho, M.P., Lambert-Torres, G., da Silva, L.E.B., da Silva, J.G.B., Neto, J.C., Bortoni, E., Lazarek, H.: Attack and Fault Identification in Electric Power Control Systems: An Approach to Improve the Security. In: Proceedings of Power Tech 2007, Lausanne, Switzerland, pp. 103–107. IEEE Press, Los Alamitos (2007)
3. Verba, J., Milvich, M.: Idaho National Laboratory Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS). In: Proceedings of the 2008 IEEE Conference on Technologies for Homeland Security, Waltham, MA, USA, pp. 469–473. IEEE Press, Los Alamitos (2008)
4. Svendsen, N.K., Wolthusen, S.D.: Modeling and Detection of Anomalies in Critical Infrastructure Networks. In: Papa, M., Shenoi, S. (eds.) Proceedings of the Second Annual IFIP WG 11.10 International Conference on Critical Infrastructure Protection. Critical Infrastructure Protection II, Arlington, VA, USA, pp. 101–107. Springer, Heidelberg (2008)
5. Watts, D.: Security & Vulnerability in Electric Power Systems. In: Proceedings of the 35 North American Power Symposium (NAPS 2003), Rolla, MO, USA, October 2003, pp. 559–566 (2003)
6. Motta Pires, P.S., Oliveira, L.A.H.G.: Security Aspects of SCADA and Corporate Network Interconnection: An Overview. In: Proceedings of the 2006 International Conference on Dependability of Computer Systems (DepCos – RELCOMEX 2006), Szklarska Proeba, Poland, pp. 127–134. IEEE Press, Los Alamitos (2006)

7. Krutz, R.L.: Securing SCADA Systems. John Wiley & Sons, New York (2006)
8. Byres, E., Hoffman, D.: The Myths and Facts behind Cyber Security Risks for Industrial Control Systems. Technical report, Department of Computer Science, University of Victoria, Victoria, BC, Canada (April 2004)
9. Gamez, D., Nadjm-Tehrani, S., Bigham, J., Balducelli, C., Burbeck, K., Chyssler, T.: Safeguarding Critical Infrastructures. In: Dependable Computing Systems: Paradigms, Performance Issues, and Applications, New York, NY, USA. John Wiley & Sons, Chichester (2005)
10. Yang, D., Usynin, A., Hines, J.W.: Anomaly-Based Intrusion Detection for SCADA Systmes. Technical report, Department of Nuclear Engineering, University of Tennessee, Knoxville, TN, USA (September 2006)
11. Cheung, S., Dutertre, B., Fong, M., Lindqvist, U., Skinner, K., Valdes, A.: Using Model-based Intrusion Detection for SCADA Networks. In: Proceedings of the SCADA Security Scientific Symposium, Miami Beach, FL, USA, January 2007, pp. 127–134 (2007)
12. Bigham, J., Gamez, D., Lu, N.: Safeguarding SCADA Systems with Anomaly Detection. In: Gorodetsky, V., Popyack, L.J., Skormin, V.A. (eds.) MMM-ACNS 2003. LNCS, vol. 2776, pp. 171–182. Springer, Heidelberg (2003)
13. Schlesser, J.E., Armstrong, D.J., Cinar, A., Ramanauskas, P., Negiz, A.: Automated Control and Monitoring of Thermal Processing Using High Temperature, Short Time Pasteurization. Journal of Dairy Science 80(10), 2291–2296 (1997)
14. Wang, X.R., Lizier, J.T., Obst, O., Prokopenko, M., Wang, P.: Spatiotemporal Anomaly Detection in Gas Monitoring Sensor Networks. In: Verdone, R. (ed.) EWSN 2008. LNCS, vol. 4913, pp. 90–105. Springer, Heidelberg (2008)
15. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press, Cambridge (2000)
16. McEvoy, T.R., Wolthusen, S.D.: Using Observations of Invariant Behavior to Detect Malicious Agency in Distributed Environments. In: Proceedings of IT Incident Management and IT Forensics (IMF 2008), Mannheim, Germany. Lecture Notes in Informatics, vol. 140, pp. 55–72. GI (2008)
17. Mouss, H., Mouss, D., Mouss, N., Sefouhi, L.: Test of Page-Hinckley: An Approach for Fault Detection in an Agro-Alimentary Production System. In: Proceedings of the 5th Asian Control Conference, Melbourne, Australia, vol. 2, pp. 815–818. IEEE Press, Los Alamitos (2004)
18. Qin, S.J., Badgwell, T.A.: An Overview of Nonlinear Model Predictive Control. In: Nnolinear Model Predictive Control, Boston, MA, USA. Birkhäuser, Basel (2000)
19. Zhao, Y., Zhou, S., Li, L.: Dynamic Characteristics Modeling of a Heat Exchanger Using Neural Network. In: Proceedings of the First International Conference on Intelligent Networks and Intelligent Systems (ICINIS 2008), Wuhan, China, pp. 13–18. IEEE Press, Los Alamitos (2008)
20. Jalili-Kharaajoo, M., Araabi, B.N.: Neural Network Based Predictive Control of a Heat Exchanger Nonlinear Process. Istanbul University Journal of Electrical & Electronics Engineering 4(2), 1219–1226 (2004)

Optimisation of Critical Infrastructure Protection: The SiVe Project on Airport Security

Marcus Breiing[1], Mara Cole[2], John D'Avanzo[3,7], Gebhard Geiger[4], Sascha Goldner[5], Andreas Kuhlmann[2], Claudia Lorenz[6], Alf Papproth[6], Erhard Petzel[1], and Oliver Schwetje[2]

[1] ckc AG, Am Alten Bahnhof 13, 38122 Braunschweig, Germany
[2] Bauhaus Luftfahrt, Boltzmannstrasse 15, 85748 Garching b. München, Germany
[3] EADS Deutschland GmbH, EADS Innovation Works, 81663 München, Germany
[4] Technische Universität München, Fakultät für Wirtschaftswissenschaften, Institut für Finanzmanagement und Kapitalmärkte, Arcisstrasse 21, 80333 München, Germany
[5] EADS Deutschland GmbH, Defence & Security, System Design Centre Germany, 85716 Unterschleißheim, Germany
[6] Fraunhofer-Anwendungszentrum für Logistiksystemplanung und Informationssysteme, Konrad-Wachsmann-Allee 1, 03046 Cottbus, Germany
[7] DMC Datenverarbeitungs- und Management-Consulting GmbH, Valentin-Linhof-Str. 8, 81829 München

Abstract. This paper outlines the scientific goals, ongoing work and first results of the SiVe research project on critical infrastructure security. The methodology is generic while pilot studies are chosen from airport security. The outline proceeds in three major steps, (1) building a threat scenario, (2) development of simulation models as scenario refinements, and (3) assessment of alternatives. Advanced techniques of systems analysis and simulation are employed to model relevant airport structures and processes as well as offences. Computer experiments are carried out to compare and optimise alternative solutions. The optimality analyses draw on approaches to quantitative risk assessment recently developed in the operational sciences. To exploit the advantages of the various techniques, an integrated simulation workbench is build up in the project.

Keywords: critical infrastructure protection, airport security, risk analysis, risk assessment, return on security investments, threat scenarios.

1 Introduction

To improve airport security is the primary task of SiVe, a research project co-funded by the German Federal Ministry of Education and Research since 2008 (SiVe = Verbesserung der **Si**cherheit von **Ve**rkehrsinfrastukturen). SiVe aims to develop an innovative methodology of risk management suitable to optimise critical infrastructure protection. Starting point is the fact that the management of airports is increasingly forced to combine high security standards and cost-efficiency in a more closely-knit and systematic way, but the measurements of the effectiveness and efficiency of risk reduction provisions poses methodological problems that, thus far, remain largely unresolved.

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 73–84, 2010.

The objective of this paper is to present the SiVe methodology. The next chapter describes the general approach including an example. In chapter 3, the models and methods we use are described in more detail. The integration of these methods into an overall IT architecture is indicated in chapter 4. A brief outlook on further development and research concludes the paper.

2 Approach

In order to tackle basic problems of quantitative risk assessment in a systematic way, innovative methods and models for secure infrastructure design and operation will be developed and tested in two distinct phases of SiVe. The two phases are:

- *Analysis, modelling and simulation of threat scenarios, their circumstantial constraints and likely consequences.* This part of the project is designed to specify and assess security risks systematically and in quantitative terms to optimise relevant parameters under cost/benefit constraints.
- *Integration of these approaches into a joint methodology and workbench which leads to an expert system for the risk management of critical infrastructures.* The integration allows the combined analysis and optimisation of all relevant security systems inherent in an infrastructure.

2.1 Elements and Steps

The approach draws upon a structural analysis of an airport. A tool, called *Scenario Builder,* was developed to generate scenarios, including all relevant elements of the airport (see section 3.1 for more details). The building and selecting of a certain scenario is the first step of the SiVe-procedure (see Figure 1). In the next step the selected scenario is modelled in greater detail (scenario refinement).

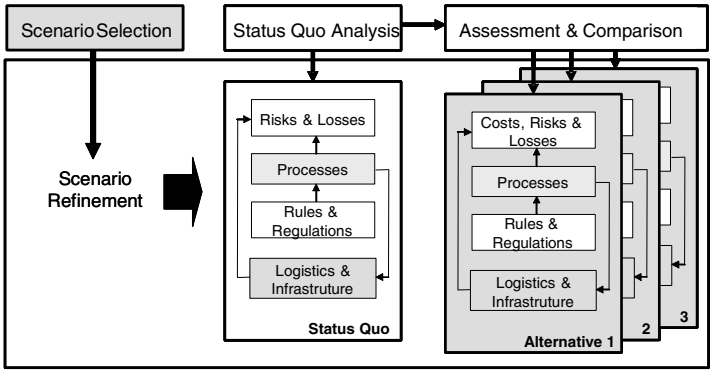


Fig. 1. Elements and steps of the SiVe approach

Figure 2 shows a seven-step model of a terrorist attack as an example of a specific scenario. All potential terrorists have to decide on practical means, an appropriate access path and how and where to enter this path. They consider alternatives and

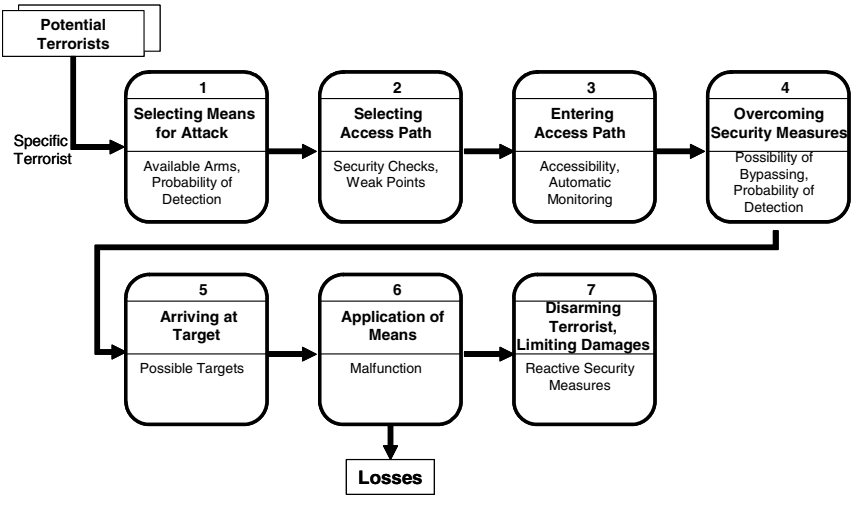


Fig. 2. A generic process model for a terrorist attack

parameters in these decisions such as available weapons, known weak points and existing monitoring systems. The possibility of bypassing and the probability of being detected are also relevant factors for the attacker. In critical infrastructure systems, terrorists are forced to overcome existing security measures before they can reach their targets. If a terrorist is able to overcome the security measures, he will apply the means he has selected. Depending on the destructiveness of the means, the losses incurred will differ. After the attack, the focus is on disarming the terrorist and limiting the damage.

Table 1. Creating a specific scenario using the generic process model

Step	Parameters/Actions
1	Goal: to kill a maximum number of passengers in the security section Means: a bomb, not detectable by metal scanners, fixed to the body.
2	Passenger handling
3	National flights, security check
4	Metal detector, security officer, quota
5	Security section, number of people
6	Ignite bomb

Because of limited space, we will use a simplified, but realistic scenario to illustrate the SiVe procedure. Table 1 shows parameters and actions for our scenario. There are two alternative cases. If a terrorist (a suicide bomber) has to pass only the metal detector gate, he can enter the security section unnoticed and ignite the bomb at a suitable moment. Alternatively, a manual check will be made, the bomb will be detected and the terrorist will ignite it directly at the security checkpoint. Because of the different conditions (number of people in the area, equipment, space) the expected loss of life will differ significantly between these two cases.

Typically metal detector gates have the function of indicating whether supplementary manual checks are necessary to ensure there is a certain quota of manual control, defined as a specific percentage of the whole flow rate of passengers. The implementation of manual controls has differing negative effects on the cost-efficiency of security checks. The higher the quota, the more personnel will be needed to conduct such controls if the flow rate is to be maintained. If new checkpoint lines have to be opened to guarantee a certain flow rate, the airport operators will be confronted with large investments in equipment and infrastructure. The adjustment of the quota is therefore a critical decision, and the optimum quota, balancing potential losses on the one hand and the cost of operations and investments on the other hand, needs to be calculated.

2.2 Refinement of Scenarios

The second step of the SiVe procedure is to refine a specific scenario by applying different models and methods. In our example, there should be rules in place to ensure a certain quota. In more complex scenarios, rules and regulations build a complex knowledge model which tells us how the system should work on a regular basis (see section 3.2).

We use such decisions as a direct input for modelling *processes* (see section 3.3). These processes take place within a certain *infrastructure*. An agent-based approach is used for the detailed modelling of logistics relevant to security (see section 3.4). In the case of a bomb attack, the modelling of the infrastructure and the dynamic of waiting queues is crucial. How many passengers will be affected by such an attack at different times of the day and in different locations?

Risks and losses depend on particular actions. Examples are the ignition of a bomb or the breakdown of security measures. Two concurrent risk events can be neutral (no cause and effect relationship) or build a cause and effect chain as risks can reinforce or weaken one another. These complex risk relationships can be modelled as a Bayesian network in order to analyse the cause and effect relationships of risks as a high-level description of the system. The main goal is to identify the most critical events which are the risk drivers. In a complex system such as an airport, there are numerous factors which influence the losses that may arise from an incident. In this case it is useful to represent risks as the probability distributions of losses (see section 3.5).

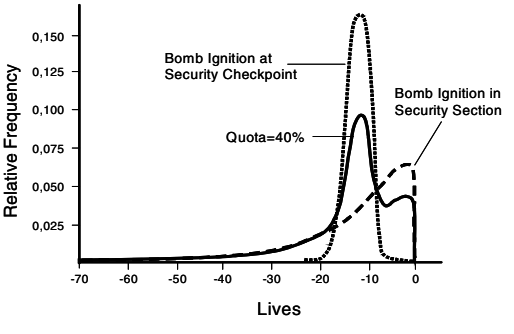


Fig. 3. Loss distributions (lives) for different cases and for a quota of 40%

For our simple example, we have computed two distributions. The first represents the relative frequency of deaths in case the terrorist ignites his bomb at the security checkpoint, the second if this happens in the security area. If the quota is zero, the second distribution will apply, if the quota is 100% the first distribution will. Quotas q between 0% and 100% are q:(1-q)-combinations of both distributions (see Figure 3).

2.3 Assessments and Comparisons

The final step in the SiVe approach is *assessing and comparing* the status quo and different alternatives while considering the costs of setting up and operating the alternatives (Figure 1). The probability distributions of losses are the input for quantitative risk assessment. This quantitative approach to risk assessment has been developed only recently in econometrics and the operational sciences and is applied in SiVe for the first time to critical infrastructure protection (see section 3.6). Figure 4 shows the results of our example. The number of lives saved, calculated as a certainty equivalent, rises with the amount of money invested. With regard to cost-efficiency, we find the maximum at a quota of approximately 46%.

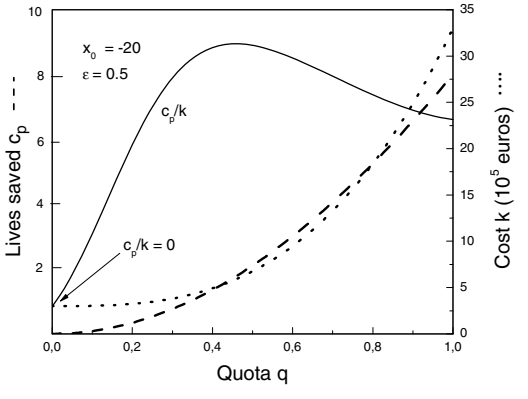


Fig. 4. Certainty equivalent c_p and cost-efficiency c_p/k as functions of the quota q, the cost k being raised to reduce the *status quo* risk p_0 to p, with k being assumed a cubic function of q as a (realistic) example. The parameter ε discounts p against p_0, as explained in [6].

3 Methods and Models

3.1 System Analysis of Airport Security

To analyse airport security systems, we have adapted a method that was originally developed to handle complexity in the field of product design [1] and for the first time used for critical infrastructure analysis. This allows us to understand the functionality of this system and to compile a systematic and exhaustive catalogue of the relevant elements.

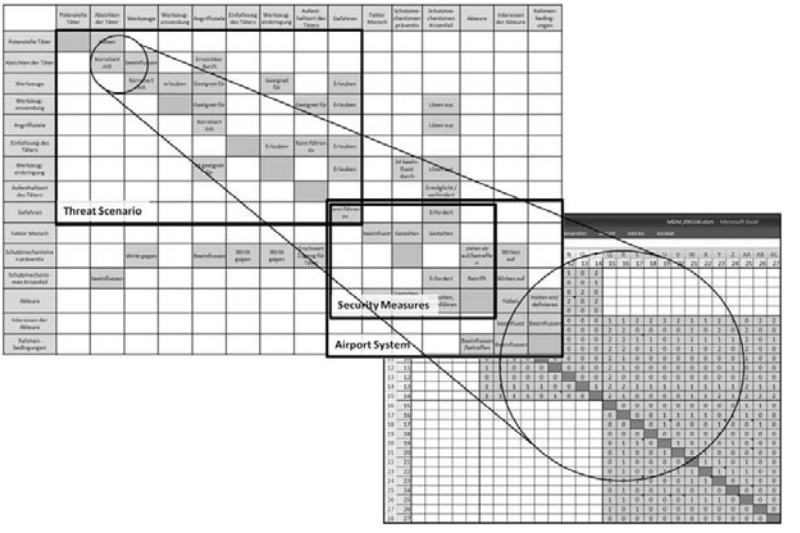


Fig. 5. Breakdown from domains to elements

At the beginning, a wide variety of elements concerning airport security and threats were assembled by an interdisciplinary team of scientists and practitioners (data gathering phase). These items were arranged in hierarchical order and organised in so called "domains" representing the highest hierarchy level[1]. Subsequently, the interconnection of the specified domains was characterised, the direction and quality of each connection discussed and named. The data were transferred to a matrix (Dependency Structure Matrix), where each matrix cell represents the influence of one domain on another and the labelling specifies the kind of dependency that applied. The diagonal matrix cells indicate self-reflexive dependencies.

The next step and main part of the data gathering was to break down the domains into the lowest hierarchy, the element level. Each filled matrix cell of the domain matrix stands for a discrete sub-matrix on the element level (element matrix) (see Figure 5). Dependencies between system elements were specified. In most cases, simple binary decisions (no influence or influence) could be made that were recorded in the element matrix by filling the cells with a "0" or "1". In more complex cases, the relationship between the elements was refined by adding a weighting with a maximum of "+2" and a minimum of "-2" indicating (strong) positive or (strong) negative influences respectively. In the final step our resulting matrix was validated by re-presenting various historical and fictional scenarios.

This approach allows us to build consistent threat scenarios based on equivalent, invariable categories. Taking the complexity of the system into account, this can be a rather complicated process. Therefore a MS Excel-based software (Scenario Builder)

[1] The domains are the following: Potential Attacker, Intention of Attacker, Tool/Weapon, Use of Tool/Weapon, Target, Approach of Attacker, Insertion of Tool/Weapon, Location of Attacker, Threat, Human Factor, Security Measure (Preventive), Security Measure (in Case of Emergency), Stakeholder, Stakeholder's Interests, General Framework.

was developed. As soon as a complete scenario is assembled, the Scenario Builder automatically lists all the elements of the airport system that are connected to the threat scenario through at least one element.

3.2 Rules and Regulations Modelling

Today rules and regulations are the basis for implementing security measures and processes in critical infrastructures. For the many different authorities who are involved (i.e. airport operators, local and federal police, different fire departments, etc.), they provide guidelines for decision making. Thereby the complexity of the system is very high. We decided to use a specific case-oriented rule-based expert system, the Erudine Behaviour Engine [2].

Rules and regulations are gathered as requirements and are modelled as a so-called *knowledge model*, which is developed by using training data, so-called *situations*. Each situation defines rules or behaviours, which can be correlated to requirements in order to track down which security requirements have been implemented and which will not. As a result, we can check the consistency of (applied) rules during the implementation. The behaviour of the overall system is obtained by gathering all the single elements. The resulting knowledge model then defines the overall behaviour of the system. In a further step, actual data can be used to validate the overall behaviour. For details see [2].

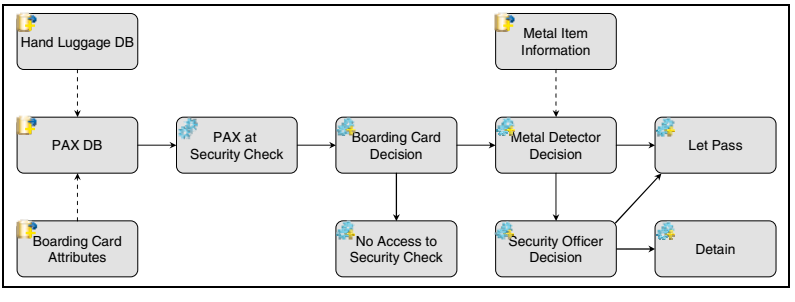


Fig. 6. Passenger screening knowledge model in the attack scenario

Within this approach, we can identify all the "rule-based" decisions that provide the framework for the scenario-based process models in the SiVe simulation framework. Figure 6 shows this in our example.

3.3 Stochastic and Process Modelling

Essential elements of critical infrastructures can be viewed as different, concurrent processes, which take place in part independently of each other. A lot of process modelling languages is available, but very few incorporate all the techniques necessary for the efficient modelling and simulation of complex systems. Important techniques are structured analysis methods for the decomposition of complex systems into process clusters, communication methods for concurrent processes such as

me-sages between processes, shared data and objects and synchronisation methods such as synchronous and asynchronous message exchange.

The Parallel Activities Specification Scheme (PASS) integrates such techniques in an orthogonal way [3]. The specification is done in three steps: (1) decomposition into process clusters, (2) decomposition of process clusters into processes, (3) description of these processes. Figure 7 shows the PASS model for our example. The terrorist process is visualised at the left hand side, which is a refinement of the generic attack process (see chapter 2). A fully specified model is directly executable with a BPEL[2] compatible workflow engine [4].

A workflow system expects inputs from users or connected systems and produces outputs to other users and systems controlled by the model process logic. These users and systems are simulated by a stochastic simulation engine as part of the SiVe engine core which will be developed during the SiVe project (see also chapter 4 for the SiVe architecture). This simulation engine will also provide all the necessary stochastic functions and generators for instances which represent the objects processed during simulation. Risk events, like the "ignition of a bomb", are modelled as special actions, which can interact with other risk events to represent the cause and effect relationships between risks. Central outputs are distributions of possible losses, which can be generated at different aggregation levels.

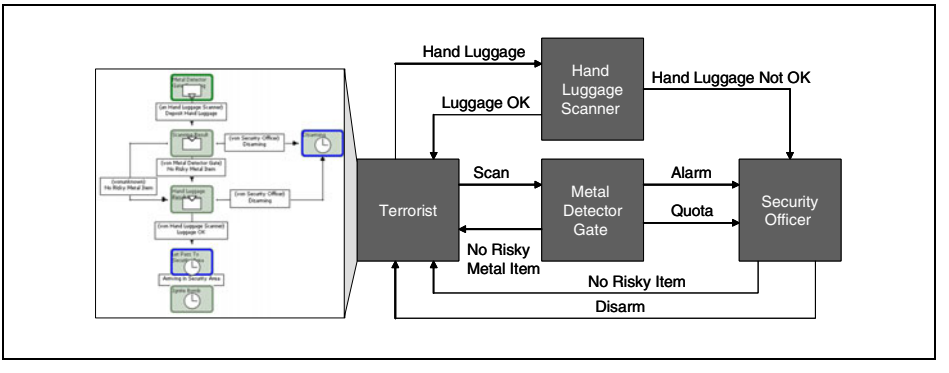


Fig. 7. PASS Model of the attack scenario

3.4 Logistic and Agent-Based Modelling

The influence of logistics and of stakeholder communication and interactions is treated in SiVe, with the use of a multi-paradigm simulation model, which combines process oriented/discrete event and agent-oriented simulation (Figure 8). The agents are represented as intelligent acting objects in a process-oriented simulation. For our scenario, the attacker movements and actions are described in a highly detailed way (e.g. walking path, running speed, destination route).

As a first example in our reference scenario, we focussed on the question of how strong the effect of a bomb explosion may be on persons present in the security

[2] Business Process Execution Language (BPEL), is an OASIS [5] standard executable language.

section or in the queue close to the security checkpoint. In a simplified approach, the agents are modelled assuming a maximum quantity of explosive of 3 kg. The agents involve a logic which triggers both the scanning process at the metal detector gate and the event „ignite bomb“. This event has a substantial influence on the nearby passengers' movement behaviour.

In addition to chronological and geometrical parameters, the event-based simulation relies on theories, methods and functions to describe the passengers' behaviour. Security clearance processes are defined in the process model (see section 3.3), while decision nodes representing rules and regulations are supported by Erudine (see section 3.2).

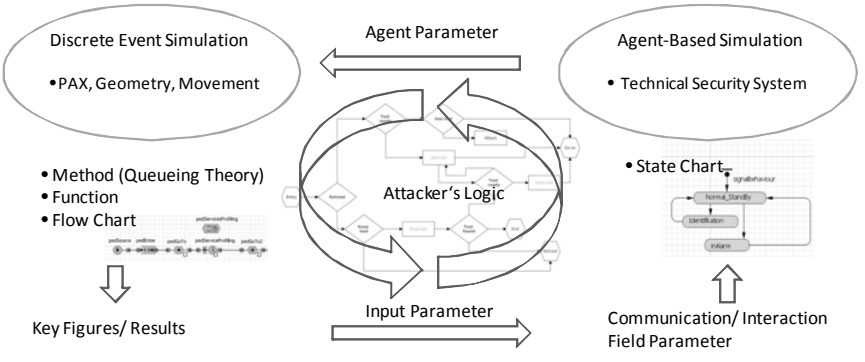


Fig. 8. Multi-paradigm approach for passenger logistics

Our process-based simulation allows us to identify bottlenecks and to meet quantitative statements in order to handle time, queue length or passage size at these bottlenecks. Defining appropriate attacker profiles and a corresponding agent logic as well as considering logistic aspects of an airport infrastructure (building aspects, number of people at different times of the day, etc.), we can calculate realistic loss distributions for quantitative risk assessments on the basis of our simulation experiments.

3.5 Quantitative Risk Assessment

Existing methods to assess risks and risk reduction measures tend to be ambiguous and controversial since they are either *ad hoc* rather than systematic or hard to operationalise. In either case, they may not provide the reliable information needed for risk-oriented decision making.

Advance has been made on the basis of recently developed econometric approaches to non-expected utility theory and the statistical foundations of quantitative risk assessment [6], [7], [8]. Within these approaches, risks have been described as random variables X with real values x and probability density functions f that are assessed by a suitable probability-dependent utility function $u(f, x)$ and the generalised utility average (or "non-expected utility")

$$U(f) = \int u(f, x)f(x)\,\mathrm{d}x.^{3}$$

Risks can be assumed to involve arbitrary kinds of loss or damage such as monetary loss, numbers of persons injured, infected or killed in an incident. Loss means $x < x_0 = 0$, where x_0 is the "aspiration level" or "neutral point of reference" which can always be normalised to zero. Suitable utility functions $u(f, x)$ have been shown to exist and are explicitly represented in terms of elementary algebraic functions under many realistic conditions [6], [7], [8]. The explicit representations of $u(f, x)$ and $U(f)$ render the present concept of risk assessment operational in each particular application. This was not possible in previous approaches to expected and non-expected utility, which often did not admit explicit algebraic representations.

A real number $c(f)$ is called the *certainty equivalent of the risk* $f(x)$ if $f(x)$ and the sure amount $c(f)$ of money (number of injured or deaths, etc) are indifferent in utility terms, that is,

$$U(f) = \int u(f, x)f(x)\mathrm{d}x = u(c(f), c(f)) = U(c(f)).$$

The certainty equivalent of a given risk can therefore be viewed as the fair price of that risk, considering that f and $c(f)$ are equal in utility. Since $u(f, x)$ is well-defined, strictly increasing in x [6], [8], and can be explicitly calculated for every probability density function f, the price c of f is determined by $c(f) = u^{-1}(f, U(f))$ where u^{-1} is the inverse function of u. In practical applications, $c(f)$ will generally have to be determined numerically for given f [6], [7], [8]. In the special case $u(f, x) \equiv x$, one has $c(f) = \mu(f)$, or "risk neutrality". However, one usually has $c(f) < \mu(f)$, meaning that the risk of large damage (negative tail of the distribution f) is felt more strongly ("risk aversion") (Figure 9).

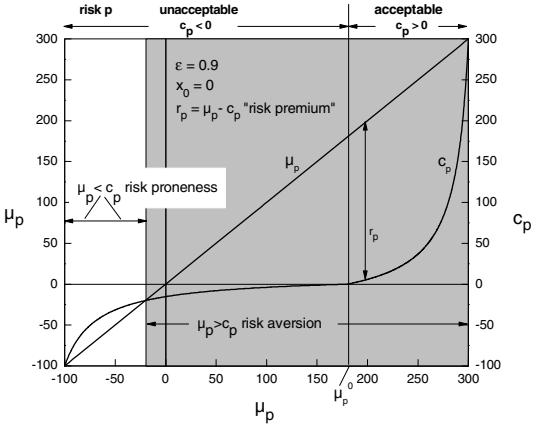


Fig. 9. Example of certainty equivalent c_p as a function of mean μ_p of a two-point distribution p according to [8]. The parameter ε is defined as in Figure 4, but different by numerical value**.**

[3] We here use continuous probability distributions $f(x)$ to characterise risks, but our formalism also applies to discrete distributions $p(x)$. [7] The examples represented in Figures 4 and 9 are two-point distributions $p(x)$.

In infrastructure security analysis, systems can be assumed to be operated with or without sufficient risk management effort. The risks f and f^* respectively attached to their operation can be estimated, considering the likely consequences of security incidents affecting any such system considered. Furthermore, the utilities $U(f)$ and $U(f^*)$ of the risks with and without risk management arrangements, respectively, can be calculated and compared. The comparison gives the effectiveness of the measures planned or taken since $U(f) \geq U(f^*)$ exactly if $c(f) \geq c(f^*)$.

Once a limit of acceptable risk f_0 has been determined [7], investments in risk management can be assessed as to whether they will reduce a given risk f^* to some residual risk f below the level of marginal acceptability f_0. However, if the limit of acceptable risk f_0 is normalised to $U(f_0) = 0$ and f^* is unacceptable $U(f^*) < 0$, one may still have $U(f^*) < U(f) < 0$. In this case, the risk management proves ineffective.

Let $k(f^*, f)$ be the cost incurred by risk managers to reduce f^* to f. The ratio $(c(f) - c(f^*))/k(f^*, f)$ gives the amount of risk reduction per euro invested. It measures the cost-efficiency of the risk reduction achieved. The price difference $c(f) - c(f^*)$ is known as the return on security investment. Risk management is optimal if f is chosen so that the cost-efficiency ratio is maximum within a given set of alternative risk mitigation choices. A numerical example is shown in Figure 4 above.

4 System Architecture of an Integrated Simulation Workbench

A challenge facing SiVe is to use the advantages of all the above-mentioned methods to build up an *integrated simulation workbench*. Integration is necessary to create a simulation model that mirrors the full complexity of critical infrastructures and presupposes specific and comprehensive system architecture.

Integration includes the development of software as well as the incorporation of existing software tools. The future architecture of SiVe will be component-based.

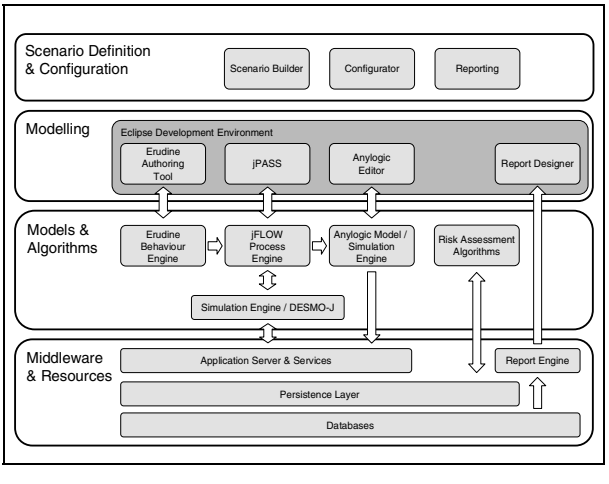


Fig. 10. SiVe high level technical architecture

On the basis of functional architecture, a technical architecture can be specified (Figure 10). The various components can be abstracted at the highest level into four categories or layers: (1) Scenario Definition & Configuration (the end-user interface); (2) Modelling (for risk and business analysts, modellers), (3) Models & Algorithms (the SiVe engine core) and (4) Middleware & Resources (the infrastructure layer).

5 Outlook

The results we presented in this paper are based on work in progress. Further activities include

- development and simulation of more complex scenarios including future security technologies such as new camera and scanner technologies to evaluate if the investment in such technologies is efficient under cost/benefit constraints;
- validation of data and estimation of the relevance of the chosen scenarios through interviews with experts;
- development of an integrated expert system based on our simulation models;
- monitoring of future and/or not yet implemented security technologies;
- ethical evaluation of security mechanisms in addition to the risk assessments.

The SiVe project will end in June 2011.

Acknowledgements

We would like to thank the Munich airport for support and the many interviews and workshops held in the course of the SiVe project. The authors are particularly grateful to the German Federal Ministry of Education and Research (BMBF) for substantial financial support to SiVe as part of the Ministry's Hightech Strategy and Security Research Programme.

References

1. Lindemann, U., Maurer, M., Braun, T.: Structural Complexity Management. An Approach for the Field of Product Design. Springer, Berlin (2009)
2. http://www.erudine.com
3. Fleischmann, A.: Distributed Systems – Software Design and Implementation, pp. 201–213. Springer, Berlin (1994)
4. http://www.jcom1.com
5. http://www.oasis-open.org
6. Geiger, G.: On the statistical foundations of non-linear utility theory. European Journal of Operational Research 136, 449–465 (2002)
7. Geiger, G.: Risk acceptance from non-linear utility theory. Journal of Risk Research 8, 225–252 (2005)
8. Geiger, G.: An axiomatic account of status quo-dependent non-expected utility: pragmatic constraints on rational choice under risk. Mathematical Social Sciences 55, 116–142 (2008)

Cyber-Critical Infrastructure Protection Using Real-Time Payload-Based Anomaly Detection*

Patrick Düssel[1], Christian Gehl[1], Pavel Laskov[1,2], Jens-Uwe Bußer[3], Christof Störmann[3], and Jan Kästner[4]

[1] Fraunhofer Institute FIRST
Intelligent Data Analysis, Berlin 12489, Germany
[2] University of Tübingen
Wilhelm-Schickard-Institute for Computer Science, Tübingen, Germany
Siemens AG
[3] Corporate Technology, Information and Communications, München, Germany
[4] Industrial Automation Systems, Research & Development, Karlsruhe, Germany

Abstract. With an increasing demand of inter-connectivity and protocol standardization modern cyber-critical infrastructures are exposed to a multitude of serious threats that may give rise to severe damage for life and assets without the implementation of proper safeguards. Thus, we propose a method that is capable to reliably detect unknown, exploit-based attacks on cyber-critical infrastructures carried out over the network. We illustrate the effectiveness of the proposed method by conducting experiments on network traffic that can be found in modern industrial control systems. Moreover, we provide results of a throughput measuring which demonstrate the real-time capabilities of our system.

1 Introduction

Industrial control systems such as supervisory control and data acquisition systems (*SCADA*), distributed control systems (*DCS*), and energy distribution systems (*EDS*) are used to monitor and control industrial automation processes and have been successfully deployed in critical infrastructures including power plants, power and water distribution, and traffic systems. Over the last decade considerable effort has been done to protect computer networks. However, a comparable small amount of research has been dedicated to cyber-security related aspects of critical infrastructure protection mainly because control systems were based on proprietary protocols and separated from public networks.

Nowadays, with the increasing demand of inter-connectivity and the ongoing convergence towards standardized network protocols communication is realized using well-known transport-layer protocols such as TCP/IP. Consequently, the

* This work was supported by the German Bundesministerium für Bildung und Forschung (BMBF) under the project ReMIND (FKZ 01-IS07007A).

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 85–97, 2010.

risk of becoming exposed to novel threats is raised. As availability is most eminent for real-time systems countermeasures to ensure integrity and confidentiality of communication can be barely applied as they usually come along with a reduction of availability. With regard to the strong utilization of computer networks and the increasing transparency of communication software patching becomes not only a very important but also a difficult task. Unfortunately, patching is not always an option since it usually requires a reboot during rare maintainance intervals or, even worse, if it cannot be guaranteed that the patch does not alter the system behavior. As a consequence, critical services in control systems remain vulnerable for a long period of time which cannot always be compensated by existing technical or administrative controls.

Thus, in order to provide adequate protection of process control networks reliable and fast intrusion detection (IDS) is crucial. Intrusion detection methods can be broadly categorized into *signature detection* and *anomaly detection*. While signature detection identifies attacks based on known attack characteristics, anomaly detection tags suspicious events by measuring a deviation from a model of normality. Signature-based intrusion detection systems possess a number of mechanisms for analyzing application-level content, ranging from simple scanning of payload for specific byte patterns, as in Snort [12], to sophisticated protocol analysis coupled with policy scripts, as in Bro [10]. Signature-based IDS typically exhibit a high detection accuracy and is therefore widely deployed as a proper compensating control in enterprise networks. However, the major drawback of signature-based IDS is their reliance on the availability of appropriate exploit signatures. Unfortunately, the rapid development of new exploits and their growing variability make keeping signatures up-to-date a daunting if not impossible task. This motivates investigation of alternative techniques, such as anomaly detection, that are in principle capable to detect unknown attacks. In our contribution we propose a method that is capable to reliably detect unknown, vulnerability-based attacks against industrial control systems that originate from both trusted and untrusted networks. We illustrate the effectiveness of the proposed method by conducting experiments on network traffic that can be typically found (*SCADA*) systems.

The paper is structured as follows. Related work on anomaly detection in SCADA networks is presented in Section 2. In Section 3 we present a topology of a modern SCADA network and briefly describe two attack scenarios that are addressed in our experiments. Details on the architecture of our anomaly detection system can be found in Section 4. Experimental evaluation on real network traffic is carried out in Section 5 which also provides a performance evaluation of our approach. Finally, conclusions are presented in Section 6.

2 Related Work

Anomaly-based IDS have traditionally focused on features derived from network and transport layer protocols. An example of such features can be found in the data mining approach of Lee and Stolfo [8], containing packet, connection

and time window features derived from IP and TCP headers. The same work has pioneered the use of "content" features that comprised selected application-level properties such as number shell prompts, number of failed login prompts, etc. deemed to be relevant for detection of specific attacks. Similar features comprising selected keywords from application layer protocols have been used by Mahoney and Chan for anomaly detection [9].

General content-based features using payload distribution of specific byte groups have been first proposed by Kruegel et al. [7] in the context of service-specific anomaly detection using separate normality models for different application layer protocols. Full distributions of byte values have been considered by Wang and Stolfo [14] and Bolzoni et al. [3], extended to models of various languages that can be defined over byte sequences, e.g. n-grams [13,11].

With regard to critical infrastructures previous work on anomaly detection has been mainly focused on physical measurement modeling which differs from our work in that we do not learn models over physical processes and also don't assume prior knowledge on protocols used to transfer measurements.

In the work of Bigham et al. [2] the authors propose to learn a *n-gram model* and an *invariant model* from data that is passed around the system. While the n-gram approach is used for the first four bytes of each data reading to determine a model of sign, decimal point position and most significant digits the latter model is used to determine linear dependencies between different data readings which are expressed as invariants. In a continuative work by Jin et al. [6] which specifically addresses anomaly detection in electricity infrastructures the authors extended the set of invariant models by a *value range model* which marks a data reading to be anomalous if its value exceeds a pre-determined threshold. Furthermore, a *bus-zero-sum* model is deployed which tests current inflow and outflow on a bus for equality. Given these models anomaly scores are finally combined in a probabilistic framework to reason about the likelihood of an anomaly given the set of trained models. Clearly, the features used for anomaly detection strongly depend on a particular domain.

A more network-centric approach is suggested by Antonio et al. [4]. They propose a distributed architecture for high-speed intrusion detection which stipulates the deployment of classification techniques to detect suspicious traffic patterns. It differs from our work in that their method requires label information to detect attacks.

3 SCADA Networks

SCADA networks are widely deployed to monitor and control processes that are essential to a modern society. Typically, a SCADA system consists of a *master terminal unit* (MTU), a *human machine interface* (HMI) and one or more *remote terminal units* (RTU) which are connected to a number of sensors and actuators in the field. Field device data is periodically transfered to the MTU which continuously reassembles an image of the overall ongoing process. The topology of a SCADA network satisfying state-of-the-art security concepts [1] is shown

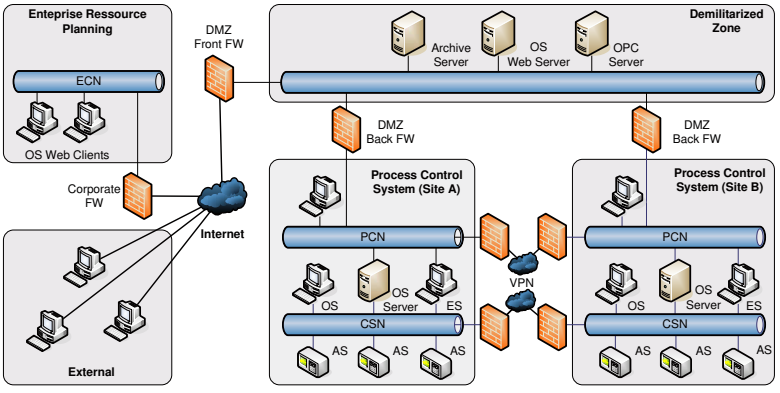


Fig. 1. General topology of a SCADA system

in Fig. 1. The network basically consists of three segments which are segregated by firewalls: an untrusted *enterprise control network* (ECN), a semi-trusted *demilitarized zone* (DMZ) and a *process control system* forming a trusted network. The process control system consists of a *control system network* (CSN) and a *process control network* (PCN). The CSN as the lowest layer network is made up of *automation stations* (AS) which are RTUs capable to exchange field device data via Industrial Ethernet to satisfy real-time requirements. The PCN which is located above the CSN constitutes the most critical part of a SCADA system. It contains a MTU referred to as *operator station* (OS) server and an *engineering station* (ES). While the OS server controls field devices attached to the CSN the engineering station provides an interface for the configuration of systems in both CSN and PCN (e.g. programming of individual AS). Communication between trusted networks is carried out via secure tunnels (e.g. VPN). The DMZ which separates untrusted and trusted networks provides a service interface to untrusted networks.

Penetration of a SCADA system requires previous exploitation of existing software vulnerabilities. Generally, there are two kinds of threat scenarios that we address in our contribution:

- ***External Threat.*** SCADA systems increasingly provide interfaces to untrusted networks such as corporate networks or the Internet. A threat agent perpetrates a device located in the DMZ (e.g. web server) from an external network either to prevent the system from being accessed by others or to carry out relayed attacks against the process control system.
- ***Internal Threat.*** A threat agent with direct access to the process control system (CSN and PCN) located at one of the facility sites mounts exploit-based attacks against critical services in the network. This threat becomes particularly serious in the presence of unmanned stations and wireless communication.

4 Methodology

The key benefit of payload-based anomaly detection lies in its ability to cope with unknown attacks. The following four stages outline the essential building blocks of our approach and will be explained in detail for the rest of this section.

1. *Network Sensor.* Inbound transport layer packets are captured from the network by *Bro*[1] which provides fast and robust TCP re-assembly. TCP Payload is extracted and forwarded to the feature extraction stage.
2. *Feature Extraction.* Byte sequences are mapped into a feature space which is defined by the set of sequential features extracted from incoming sequences. The utilization of efficient data structures allows to operate in high-dimensional feature spaces. Details on the feature extraction process can be found in Section 4.2.
3. *Similarity Computation.* A proper definition of similarity between byte sequences is crucial for payload-based anomaly detection. The similarity between byte sequences is determined by computing the pairwise distance between their respective vectorial representation. The similarity computation is explained in Section 4.3.
4. *Anomaly Detection.* The anomaly detector initially learns a global model of "normality" which is represented by the center of mass of training data points. At detection time arriving byte sequences are compared to the previously learned model and based on a distance an anomaly score is calculated. The anomaly detection process is described in Section 4.4.

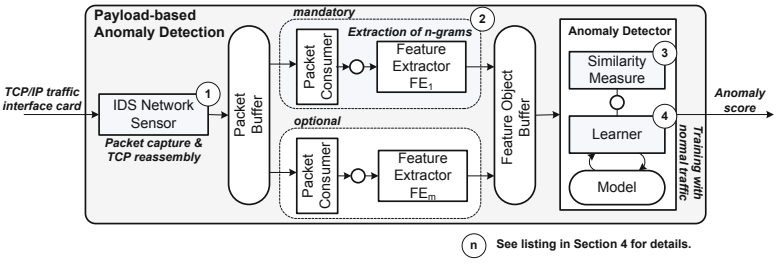


Fig. 2. Payload-based anomaly detection system

4.1 Network Sensor

Bro as the basis of our prototypical implementation is a Unix-based Network Intrusion Detection System that passively monitors network traffic. To match our requirements as a TCP/IP network sensor we exclude parsing of network traffic and signature matching. The Berkeley Packet Filters (BPF), also known as tcpdump expressions, provided by *Bro* can be used in a highly flexible and

[1] See http://www.bro-ids.org/ for details.

adapted mode to process the desired packets for detection. Moreover, *Bro* takes care of fragmented packets and TCP re-assembly which are important factors for robust processing of TCP network traffic. Byte sequences are extracted from the payload of incoming packets and forwarded to the feature extraction stage.

4.2 Feature Extraction

Anomaly detection usually requires data to be in a vectorial representation. Therefore, the feature extraction process embeds a byte sequence s into a feature space $\mathcal{F}$ in which similarity between sequences can be computed. By moving a sliding window of a particular length n over a sequence s a set of unique, sequential features – so called n-grams – is extracted. The resulting feature space is defined over the set $I \subseteq \Sigma^n$ of possible n-grams u induced by an alphabet Σ:

$$\phi(s) \mapsto (\phi_u(s))_{u \in I} \in \mathcal{F}, \quad u \in \Sigma^n \tag{1}$$

Once a sequence is embedded into $\mathcal{F}$ various feature maps ϕ can be applied. Commonly used feature maps for *contiguous strings* are explained below:

- *Count Embedding.* The value of the coordinate $\phi_u^{cnt}(s)$ reflects the count of a string u contained in s.
- *Frequency Embedding.* The value of the coordinate $\phi_u^{freq}(s)$ reflects the term frequency of a string u contained in s. Essentially, this mapping corresponds to a *count embedding* normalized by the maximum number of n-grams contained in s.
- *Binary Embedding.* The value of the coordinate $\phi_u^{bin}(s)$ reflects the presence of a string u in s.

4.3 Similarity Measure

The utilization of a geometric representation of a byte sequence through ϕ allows to deploy classical, vector-based similarity measures such as distance functions. A list of relevant function is provided in Table 1.

Table 1. Similarity measures

Name	Similarity	Name	Similarity
Manhattan	$\sum_{i=1}^{n} \lvert x_i - y_i \rvert$	Canberra	$\sum_{i=1}^{n} \frac{\lvert x_i - y_i \rvert}{x_i + y_i}$
Euclidean	$\sum_{i=1}^{n} (x_i - y_i)^2$	Chi-Squared	$\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i}$

4.4 Anomaly Detection

Anomalies can be considered as deviations from a previously learned model of normality which is represented as the center of mass of a set of data points. To this end, the center of mass $\mathbf{c}$ can be defined by:

$$\mathbf{c} = \frac{1}{\ell} \sum_{i=1}^{\ell} \phi(x_i),\ \phi(x_i) \in \mathbb{R}^n \tag{2}$$

where $\phi(x_i)$ refers to a training point explicitly embedded in a n-dimensional geometric space as described in Section 4.2. In order to down-weight sparse representations $\mathbf{c}$ is normalized as follows:

$$\hat{\mathbf{c}} = \mathbf{c}^T(\mathbf{\Sigma^D W}^{-1}), \tag{3}$$

where $\mathbf{\Sigma^D} = diag(w)$ denotes the standard deviation of individual dimensions in $\mathcal{F}$ observed over ℓ training points and $\mathbf{W} = diag(\|\mathbf{w}\|_1 I_{1\times n})$ refers to the 1-norm of w. Finally, an anomaly score S_x for an unknown data point x is computed by calculating the distance to the center of mass of training data:

$$S_x = \sum_{j=1}^{n} d(\hat{\mathbf{c}}_j, \phi_j(x)) \tag{4}$$

A similar anomaly detection method is referred to as *Payl* [14] which relies on the computation of a *simplified Mahalanobis distance*. The anomaly score S_x is defined as the variance-scaled distance from an unknown data point x to the center of mass $\mathbf{c}$:

$$S_x = \sum_{j=1}^{n} \frac{d(\mathbf{c_j}, \phi_j(x))}{\sigma_j} \tag{5}$$

5 Experiments

With regard to the attack scenarios outlined in Section 3 we evaluate our method on two data sets containing traffic monitored at our institute as well as in a SCADA testbed. For each of the two data sets we recorded a collection of service-specific attacks including buffer overflows which can be used for *privilege escalation* as well as web application attacks. Exploits were taken from the Metasploit framework[2] and from common security forums such as *securityfocus.com* and *remote-exploit.org*.

The first data set (***Web07***) contains inbound HTTP traffic captured in a DMZ over a period of five days and consists of approx. 1,000,000 TCP packets. The corresponding attack set comprises 42 attack instances exploiting 11 different vulnerabilities in HTTP services together with a *Nessus* vulnerability scan. The list of HTTP-related exploits can be found in Table 2.

The second data set ***Aut09*** which comprises approx. 765,103 TCP packets was captured in a process control network and contains payload of binary application-layer protocols such as RPC/DCOM, SMB and Netbios.

[2] http://www.metasploit.com

Table 2. Attack sets ([1] *Code Red*, [2] *W32.Blaster*, [3] *Conficker*)

	Id	CVE	Description	Type	Id	CVE	Description	Type
HTTP	1	2002-0071	IIS (ISAPI/HTR Script)	Buf	7	2006-5478	Novell eDirectory	Buf
	2	2001-0500[1]	IIS (ISAPI/Indexing)	Buf	8	2003-1192	IA WebMail	Buf
	3	2001-0241	IIS (ISAPI/Printing)	Buf	9	2000-0884	IIS Dir. Traversal	Web
	4	2004-1134	IIS (ISAPI/W3Who)	Buf	10	2006-2644	AwStats (Logging)	Web
	5	2003-0109	IIS (NTDLL/WebDAV)	Buf	11	2005-2847	Barracuda (Spam)	Web
	6	2002-2270	Alt-N WebAdmin	Buf	12	-	Nessus scan	Web
RPC	1	2003-0352[2]	RPC (DCOM)	Buf	6	2006-4696	SMB Mailslot	Buf
	2	2003-0533	LSASS	Buf	7	2006-3441	RPC (DNS)	Buf
	3	2004-1080	RPC (WINS)	Buf	8	2008-4250[3]	SMB (SRVSVC)	Buf
	4	2005-0059	RPC (MSMQ)	Buf	9	-	RPC scan	-
	5	2006-3439	SMB (SRVSVC)	Buf				

The corresponding attack set includes 19 attack instances exploiting eight different vulnerabilities. Attacks were carried out using various attack payloads (i.e. account creation, (reverse) shell binding and VNC server injection. RPC-related exploit details are provided in Table 2.

5.1 Experimental Setup

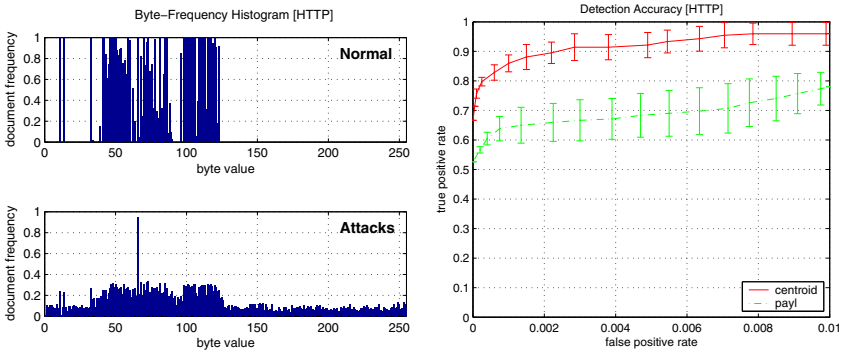
In order to find a model that maximizes the detection performance a validation phase precedes the actual evaluation of our method. It is important to mention that *data used during validation is not employed during evaluation.* To this end, data is split into three distinct partitions for training, validation and testing from which samples are randomly drawn. Each sample consists of 1000 TCP payloads. Model selection is implemented as a 10-fold cross validation. Both, validation and test samples are equally mixed with **distinct** subsets of available attack classes. Each model is trained on a training sample and subsequently validated on ten distinct validation samples. Finally, the model that maximizes the average detection accuracy during validation is applied on a test sample. The detection accuracy is measured in terms of area under *receiver operating characteristic* curve ($AUC_{0.01}$) which integrates true positive values over the false positive interval [0,0.01]. For statistical reasons experiments on both data sets are repeated over 20 repetitions.

5.2 Experiments on HTTP Traffic

In this experiment we investigate the detection of overflow-based attacks and web application attacks carried out over the well-known HTTP protocol. As shown in Table 3 cross validation reveals that variance-weighted Manhattan distance over 3-grams is chosen to be the best model over all repetitions. Considering the average AUC values for the best three models the choice of the feature embedding has only a minor impact on the detection accuracy. As shown in

Table 3. $AUC_{0.01}$ values of selected models in validation phase

Measure	Weighting	n-gram	Embedding	AUC_{avg}	AUC_{std}	$\|Model_{best}\|$
Manhattan	variance	3	freq	0.9231	±0.0370	10
Manhattan	variance	4	freq	0.9169	±0.0406	2
Canberra	none	4	bin	0.9091	±0.0395	5



(a) Byte histogram (b) Detection Accuracy

Fig. 3. Detection of unknown attacks in HTTP traffic

Table 4. False positive rate per HTTP attack class on test data

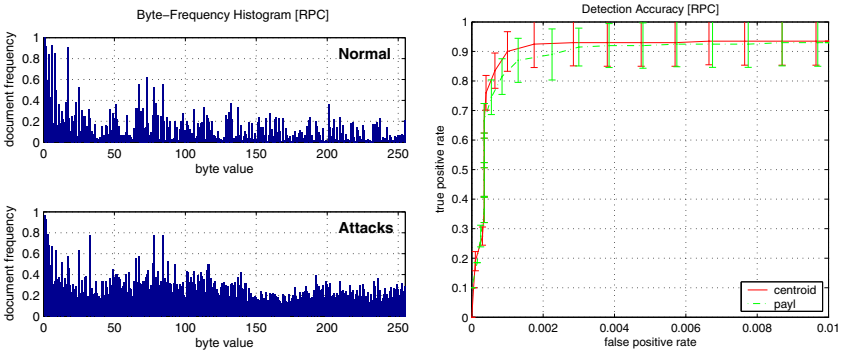
Class	Instances	Centroid		Payl	
		FP_{avg}	FP_{std}	FP_{avg}	FP_{std}
1 - 8	25	0.0000	±0.0000	0.0000	±0.0000
9	3	0.0008	±0.0013	0.0060	±0.0014
10	5	0.0072	±0.0079	0.0360	±0.0128
11	6	0.0045	±0.0053	0.0260	±0.0081
12	3	0.0056	±0.0037	0.0264	±0.0075

Fig. 3(b) with a detection rate of 88% at a false alarm rate of 0.2% our method strongly outperforms *Payl*. A detailed analysis of the cost associated with the detection of individual attack classes is given in Table 4. It shows that overflow-based attacks are perfectly detected while some web application attacks suffer from a small number of false positives.

This is not a surprise considering the significant differences in the byte histograms of normal data and overflow-based attacks which are compared in Fig. 3(a). Unlike overflow-based attacks which heavily rely on the utilization of bytes in the range between **0x7f** and **0xff** to carry malicious opcode web application attacks usually exploit vulnerabilities of scripts using bytes that are commonly found in normal HTTP traffic. This explains the comparably high cost associated with the detection of this type of attacks. However, the detection

Table 5. $AUC_{0.01}$ values of selected models during validation

Measure	Weighting	n-gram	Embedding	AUC_{avg}	AUC_{std}	$\|Model_{best}\|$
Chi-Squared	variance	4	freq	0.8655	±0.0721	13
Canberra	none	4	bin	0.8548	±0.0892	3
Canberra	none	2	bin	0.8459	±0.0911	3



(a) Byte histogram (b) Detection Accuracy

Fig. 4. Detection of unknown attacks in RPC/SMB traffic

accuracy for web application attacks can be further enhanced by incorporating syntactic information into sequential feature representations [5].

5.3 Experiments on RPC Traffic

While in the previous section we address the problem of detecting attacks carried out over text-based application protocols such as HTTP in this section we investigate the detection of overflow-based attacks carried out over binary protocols such as Netbios, SMB and RPC. Experiments were applied to the ***Aut09*** data set. As shown in Table 5 cross validation chooses a variance-weighted Chi-squared distance using a frequency embedding as the best model.

Similarly to the results of experiments on HTTP in this experiment our method attains a detection rate of 92% at a false positive rate of 0.2%. Interestingly, although a rather complex model (i.e. Chi-squared distance over 4-grams) is chosen our method demonstrates a only marginal improvement in detection accuracy compared to the results obtained using *Payl*. However, while most of the attacks are perfectly separable from the normal data (cf. Table 6) "SMB-Mailslot" is the only attack that suffers a comparably high false positive rate of approx. 5%. This particular denial of service attack exploits a vulnerability in the Microsoft server service (SVR.SYS) and is triggered by a specially crafted but fairly short malformed SMB packet.

Table 6. False positive rate per RPC/SMB attack class on test data

Class	Instances	Centroid		Payl	
		FP_{avg}	FP_{std}	FP_{avg}	FP_{std}
1	3	0.0004	±0.0005	0.0004	±0.0005
2	1	0.0003	±0.0006	0.0003	±0.0006
3	1	0.0001	±0.0003	0.0003	±0.0005
4	1	0.0000	±0.0000	0.0000	±0.0000
5	6	0.0003	±0.0005	0.0003	±0.0005
6	1	0.0705	±0.0684	0.2783	±0.0464
7	1	0.0005	±0.0008	0.0000	±0.0000
8	4	0.0005	±0.0008	0.0018	±0.0014
9	1	0.0950	±0.1118	0.0231	±0.0097

5.4 Performance Evaluation

The performance of our prototypical implementation is tested inside of a virtual network which consists of two client machines[3] which operates as sender and receiver of network traffic. Each client is installed on a separate host[4]. To simulate a proper online scenario we use the tcpreplay tool suite[5] to replay the captured tcpdump files between both clients. In order to maintain the original transport-layer characteristics (i.e. to preserve the TCP connection on the receiver site) the Media Access Control address of all packets are rewritten to the receivers's interface card. Thus, the prototype can process full TCP connections, as it is required in a production environment. Table 7 shows the throughput in MBits per second obtained from running our prototype in single CPU mode. During measurement anomaly detection is applied to incoming traffic. The performance evaluation reveals that the throughput of our prototype depends on the ratio of inbound packets as well as the n-gram size at hand. The drop of performance linked with the increasing n-gram size is due to the fact that the dimensionality of the underlying feature space increases exponentially which also affects the feature extraction process.

Table 7. Throughput of the single CPU prototype analyzing incoming traffic

Data set	Replayed packets	Analyzed packets	1-gram	2-gram	3-gram	4-gram
Web07	1,000,000	6.0%	429.1Mbps	348.6Mbps	309.9Mbps	253.1Mbps
Aut09	765,103	52.3%	197.5Mbps	113.1Mbps	31.4Mbps	18.3Mbps

[3] FreeBSD images using 2 CPUs with 4 GByte RAM memory. The BPF buffer size is set to 10MByte.
[4] The server hardware are two Sun Fire X4100 M2 with 4 CPU x 2,8Ghz.
[5] See http://tcpreplay.synfin.net/trac/ for details.

6 Conclusion

In this contribution we propose an effective real-time payload-based anomaly detection system which can be deployed to protect critical information infrastructure assets by providing reliable protocol-independent zero-day attack detection. Our method relies on the computation of similarity between transport-layer packet payloads embedded in a geometric space. We carry out comprehensive experiments on network traffic captured in different network segments of a SCADA testbed including communication of text-based and binary application-layer protocols. With a detection rate of **88%-92%** at a false positive level of **0.2**% the method has been proved to be useful for the detection of unknown attacks in network traffic. Future work should address further improvements of detection accuracy in terms of false positive reduction by utilizing more sophisticated detection methods such as Support Vector Machines. Moreover, considering the nature of critical information infrastructures the paradigm of automatic intrusion prevention becomes indispensable. Therefore, extensions of our method should be investigated and thoroughly evaluated for utilizability in critical information infrastructures.

References

1. Security concept pcs6 and wincc-basic document. White paper, Siemens AG, A5E02128732-01 (April 2008)
2. Bigham, J., Gamez, D., Lu, N.: Safeguarding scada systems with anomaly detection. In: Gorodetsky, V., Popyack, L.J., Skormin, V.A. (eds.) MMM-ACNS 2003. LNCS, vol. 2776, pp. 171–182. Springer, Heidelberg (2003)
3. Bolzoni, D., Zambon, E., Etalle, S., Hartel, P.H.: Poseidon: a 2-tier anomaly-based network intrusion detection system. In: 4th IEEE Int. Information Assurance Workshop (IWIA 2006), pp. 144–156 (2006)
4. D'Antonio, S., Oliviero, F., Setola, R.: High-speed intrusion detection in support of critical infrastructure protection. In: Proc. 1st International Workshop on Critical Information Infrastructures Security (2006)
5. Düssel, P., Gehl, C., Laskov, P., Rieck, K.: Incorporation of application layer protocol syntax into anomaly detection. In: Sekar, R., Pujari, A.K. (eds.) ICISS 2008. LNCS, vol. 5352, pp. 188–202. Springer, Heidelberg (2008)
6. Jin, X., Bigham, J., Rodaway, J., Gamez, D., Phillips, C.: Anomaly detection in electricity cyber infrastructures. In: Proceedings of the International Workshop on Complex Networks and Infrastructure Protection, CNIP 2006 (2006)
7. Kruegel, C., Toth, T., Kirda, E.: Service specific anomaly detection for network intrusion detection. In: Proc. of ACM Symposium on Applied Computing, pp. 201–208 (2002)
8. Lee, W., Stolfo, S.: A framework for constructing features and models for intrusion detection systems. ACM Transactions on Information Systems Security 3, 227–261 (2000)
9. Mahoney, M., Chan, P.: Learning nonstationary models of normal network traffic for detecting novel attacks. In: Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 376–385 (2002)

10. Paxson, V.: Bro: a system for detecting network intruders in real-time. In: Proc. of USENIX Security Symposium, pp. 31–51 (1998)
11. Rieck, K., Laskov, P.: Language models for detection of unknown attacks in network traffic. Journal in Computer Virology 2(4), 243–256 (2007)
12. Roesch, M.: Snort: Lightweight intrusion detection for networks. In: Proc. of USENIX Large Installation System Administration Conference LISA, pp. 229–238 (1999)
13. Wang, K., Parekh, J., Stolfo, S.: Anagram: A content anomaly detector resistant to mimicry attack. In: Zamboni, D., Krügel, C. (eds.) RAID 2006. LNCS, vol. 4219, pp. 226–248. Springer, Heidelberg (2006)
14. Wang, K., Stolfo, S.: Anomalous payload-based network intrusion detection. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 203–222. Springer, Heidelberg (2004)

Decision Aid Tool and Ontology-Based Reasoning for Critical Infrastructure Vulnerabilities and Threats Analysis

Michał Choraś[1,2], Adam Flizikowski[1,2], Rafał Kozik[2], and Witold Hołubowicz[1,3]

[1] ITTI Ltd., Poznań
michal.choras@itti.com.pl
[2] Institute of Telecommunications, UT&LS Bydgoszcz
chorasm@utp.edu.pl
[3] Adam Mickiewicz University, Poznań
holubowicz@amu.edu.pl

Abstract. In this paper, a decision aid tool (DAT) for Critical Infrastructure threats analysis and ranking is presented. We propose the ontology-based approach that provides classification, relationships and reasoning about vulnerabilities and threats of the critical infrastructures. Our approach is a part of research within INSPIRE project for increasing security and protection through infrastructure resilience.

1 Introduction

The increasing success of information and communication technologies, together with the progressive disuse of dedicated communication networks are introducing a new way of controlling and managing critical infrastructures, which are currently organized as strictly connected, albeit different, elements of a single system rather than as autonomous entities to be appropriately integrated. More precisely, systems controlling critical infrastructures are rapidly moving from dedicated and proprietary solutions towards IP-based integrated frameworks made of off-the-shelf products.

SCADA is an acronym for Supervisory Control and Data Acquisition. SCADA systems are used to monitor and control a plant or equipment in industries or critical infrastructures such as water and waste control, energy, oil and gas refining and transportation. These systems encompass the transfer of data between a SCADA central host computer and a number of Remote Terminal Units (RTUs) and/or Programmable Logic Controllers (PLCs), and the central host and the supervising operator terminals. In the last years, one can observe a change in relation of dedicated SCADA systems with Information Systems. In particular, SCADA systems are now often interconnected with open information systems, via open heterogeneous WAN networks. This was not the case, when SCADA systems were firstly created: then SCADA were separate, dedicated systems with little interconnection to non-SCADA information systems [1].

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 98–110, 2010.

Hereby, we propose an ontology approach to be applied for SCADA-ICT Interdependencies, Vulnerabilities and Threats Analysis. Moreover, we will present our new approach that includes the use of ontology-based reasoning and an embedded inference engine in order to design and develop the INSPIRE Decision Aid Tool (DAT).

The paper is structured as follows: in Section 2 INSPIRE project (objectives, research challenges) is presented. The motivation and description of the ontology-based approach are given in sections 3.1 and 3.2, respectively. In section 4, the decision-aid reasoning tool is described.

2 INSPIRE Approach to Critical Infrastructures Protection

Critical Infrastructures vulnerabilities and their interdependencies analysis, especially these concerning SCADA systems and their interconnections with open telecommunications networks is a part of the ongoing research and activities in the INSPIRE project.

INSPIRE is a 2-year STREP EU Project within the Joint Call FP7-ICT-SEC-2007-1 (Critical Infrastructure Protection). The project started in November 2008 and will end in October 2010. INSPIRE research challenges are as follows:

- Analysis and modeling of dependencies between critical infrastructures and underlying communication networks;
- Designing and implementing traffic engineering algorithms to provide SCADA traffic with quantitative guarantees;
- Exploiting peer-to-peer overlay routing mechanisms for improving the resilience of SCADA systems;
- Defining a self-reconfigurable architecture for SCADA systems;
- Development of diagnosis and recovery techniques for SCADA systems;

INSPIRE project aims at developing innovative automated reconfiguration techniques such as P2P (Peer to Peer) and multi-agent reconfigurators. In particular the INSPIRE project aims at investigating the characteristics of P2P for the purpose of hardening SCADA systems against a cyber-attack. For example, in case real-time message delivery constraints are not being met (due, for example, to a denial of service attack), a P2P overlay network could be used to route message floods in an effort to ensure delivery [2].

Moreover, the use of routing engineering algorithms such as MPLS to prioritize flows is investigated. While QoS architectures employ scheduling mechanisms to differentiate flows locally at each router, our approach will be to differentiate flows by routing them along different explicit paths (MPLS networks) [2].

The above mentioned techniques will be supported by advanced diagnosis mechanism and ontology-based reasoning (INSPIRE Decision Aid Tool).

INSPIRE aims at making advancements in the definition of a distributed diagnosis and reconfiguration framework. A diagnostic system based on the concept of threshold must be able to understand the nature of errors occurring in the

system, judge whether and when some action is necessary, and finally trigger the recovery/reconfiguration/repair mechanisms to perform the adequate actions.

From the application point of view, the INSPIRE Project focuses mainly on telecommunication, energy and transportation sectors.

The INSPIRE project also establishes international links within Critical Infrastructures protection community, tightly cooperating with e.g. US GridStat Project (FP7 INSPIRE - International Cooperation).

3 Ontology-Based Description of SCADA Interdependencies, Vulnerabilities and Threats

3.1 Motivation for Ontology-Based Approach

Now in the third generation (*networked*) of SCADA systems and their functionalities are distributed across WAN, not only across LAN. The internet Protocol (IP) is used for communication between master stations and RTUs. The major positive aspect of interconnecting SCADA with WAN is disaster survivability. It means that on the basis of WAN networking and using separate locations, it is possible to build SCADA systems that can overcome the loss of one of its parts. The negative aspect of described evolution is the influence of IP-based vulnerabilities and attacks.

SCADA systems have their own vulnerabilities due to their architectures, devices, software, and due to their special protocols. However, the increasing development of communication systems and the integration of IT systems with SCADA make the latter vulnerable to cyber attacks coming from IT systems.

An ontology-based description provides a possibility to show all these connections and dependencies among SCADA systems and their security aspects. Such ontology should provide information about vulnerabilities that SCADA have and with which architectural components they are connected. Relationships and properties should show what threats and attacks may be caused by a particular vulnerability. A dedicated ontology seems to be a good tool/instrument to describe these usually complicated relationships.

Today many systems are monitored using the infrastructure of the corporate Local Area Network (LAN) / Wide Area Network (WAN), secure tunnels on public networks and, mainly in power electricity industry, particular communication technologies like power line communication (PLC); wireless technologies (including satellite) are now being widely deployed for purposes of monitoring. We also observe an increasing integration of Wireless Sensor Networks (WSN) for fine-grained monitoring and mobile devices as information/diagnosis supplies for technical stuff. New communication standards such as ZigBee, WLAN and Bluetooth are also emerging in SCADA and PCS systems. Moreover, more and more often, it is tried to guarantee the resilience of SCADA communication system by implementing a heterogeneous network, formed by the integration of two, or more, of the network technologies.

These changes have exposed SCADA systems to vulnerabilities that have already threatened standard and general purpose ICT system. Examples include

the switch from using leased telecommunications lines to shared communication infrastructures and a public network, as well as the interconnection of SCADA system with the company network and systems (due to administrative purposes, for example to facilitate billing or production forecast activity). Such measures have introduced many new vulnerabilities to SCADA systems, dramatically increasing the risk of attacks both from the internal side and from the Internet. The shared communications infrastructure thus becomes an obvious target for disrupting a SCADA network.

While physical security of critical infrastructure components (included the control system) has been already investigated, scarce attention has been paid so far to the analysis of vulnerabilities resulting from the use of commercial communication networks to transport management information between ICT systems devoted to the control of critical infrastructures. Therefore, in the context of the INSPIRE Project, ontology based reasoning about Critical Infrastructures and SCADA vulnerabilities is required.

Ontologies represent an effective tool to describe complex and interdependent symptoms related to faults and attacks, since they are meant to provide formal specification of concepts and their interrelationships. The ontology based hierarchical organization of event patterns can be used to partially automate the process of deriving queries for resiliency analysis.

3.2 INSPIRE Ontology Description

In this section our ontology approach with special consideration and emphasis on vulnerabilities is presented. Ontology has been created in OWL-DL language using the Protege 3.4 application.

The SCADA ontology aims at addressing the following key SCADA issues:

- What are vulnerabilities of SCADA components, architecture or protocols?
- How these vulnerabilities affect SCADA resources?
- Which threats/attacks may occur and what damage can they cause?

According to ISO/IEC 13335-1:2004 standard [6] vulnerabilities are considered as a property of a network security system. In this approach, SCADA resources and components have weak points named vulnerabilities. These vulnerabilities can be exploited by threats, leading to attacks. Such security system is depicted into a form of classification with properties and relationships among security issues. Main concepts, which compose main classes of proposed ontology on the basis of the ISO/IEC 13335-1:2004 standard are:

- Assets (anything that has value to the organization)
- Vulnerabilities (include weaknesses of an asset or group of assets which can be exploited by threats)
- Threats (potential cause of an unwanted incident which may result in harm to a system or organization)
- Source of attacks
- Safeguards (practices, procedures or mechanisms that reduce vulnerabilities)

These classes are connected to each other by properties. Properties can show relations, dependence of one class on another or can represent some attributes. The Vulnerabilities class properties are as follows:

- are exploited by (Threats)
- category CWE (String)
- CPE name (String)
- CVE id (String)
- CVSS access complexity (String: Low, Medium, High, Insufficient Information)
- CVSS access vector (String)
- CVSS authentication (String)
- CVSS availability (String)
- expose asset (Asset)
- have safeguard (Safeguard)
- have attack vector (String)
- have risk level (String: Low, Moderate, Height, Critical)

The properties allow to model interconnection between particular Vulnerabilities and between Vulnerabilities and Assets. The Assets class properties are:

- has vulnerability (Vulnerability)
- depends on (Application)
- runs on (Operating System)
- installed on (Hardware)
- connected to (Network)

The Safeguards properties allows to model relation between Vulnerabilities and Safeguards. The properties of this class are:

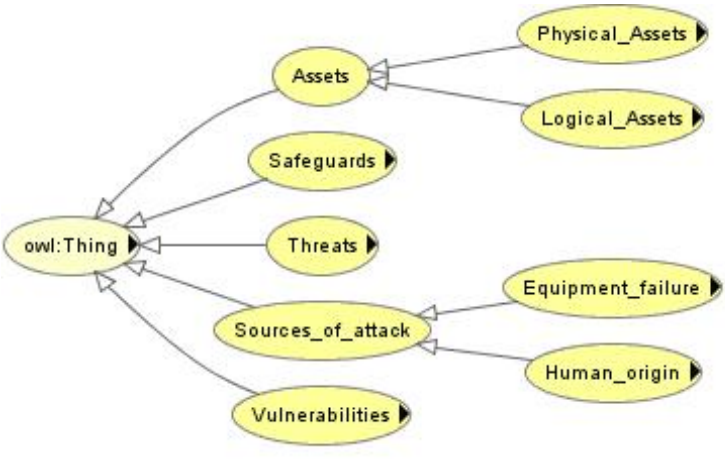


Fig. 1. Ontology: Class hierarchy - general view

– reduces (Vulnerability)
– name

The Threat class properties are:

– is caused by (Source of attack)
– is related to (Asset)
– exploits (Vulnerability)

The proposed hierarchy of ontology classes is presented in Figure 1.

Classification of key class of ontology - *Vulnerabilities*, has been based on vulnerability description given in [8] and vulnerability identification given in INSPIRE Deliverable D.2.2. [9]. Thus SCADA vulnerabilities are grouped into five sub-classes of *Vulnerabilities* class. Particular individuals are related to particular resources, applications, operating systems or protocols.

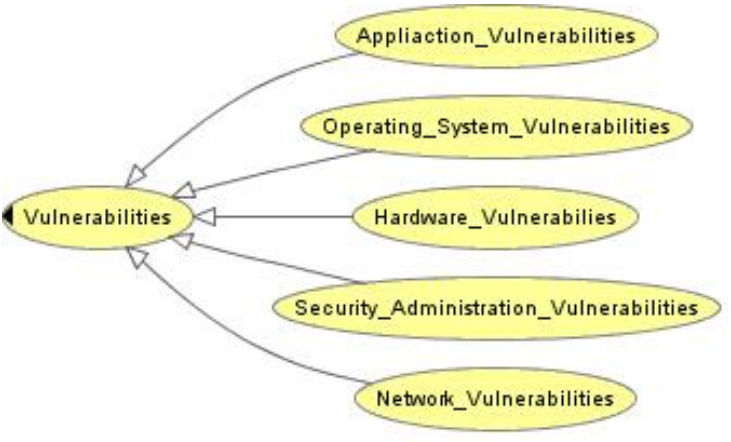


Fig. 2. Proposed vulnerabilities classification

Hierarchy of vulnerabilities classes is presented in Figure 2. Vulnerabilities derived from assets, and classification of assets is presented in Figure 3. SCADA systems created as industry systems have their own dedicated communication protocols like Modbus, Profibus or DNP3. These dedicated protocols also have vulnerabilities characteristic for them, and they are included in assets classification.

The main classes, the concepts of vulnerability classification, should be instantiated with individuals. Particular vulnerabilities related to SCADA systems in the described ontology are based on the identification of vulnerabilities performed and reported within the INSPIRE Project [9]. They are instantiated as individuals in proposed ontology. Moreover, to improve our ontology and its further usage we have also applied additional rules in SWRL Language [10].

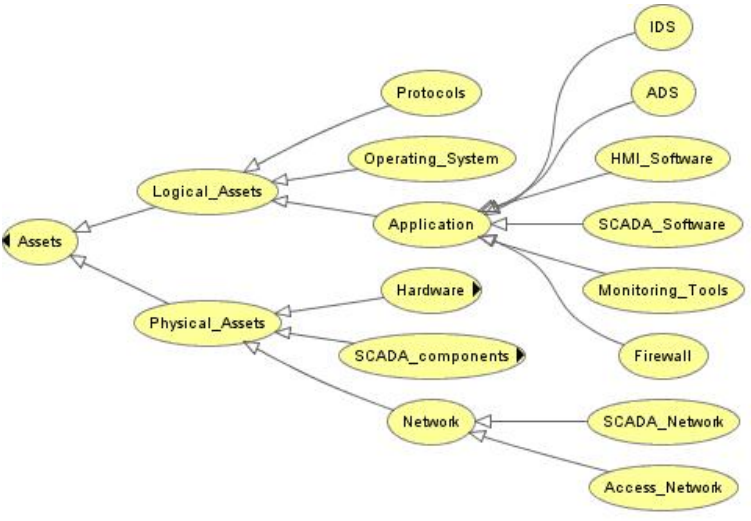


Fig. 3. Assets of SCADA Ontology

4 INSPIRE Decision Aid-Tool

4.1 Decision Support Systems for CIP

Decision Support Systems (DSS) are group of information systems that support human in different decision-making activities. The DSS applications are successfully and widely used in industry and critical infrastructure protection (CIP).

In 1987 Texas Instruments company has released GADS (Gate Assignment Display System) decision support system for United Airlines. As a result the travels delays have been reduced significantly. The system was aiding the management of ground operations at various airports. Another good example of successfully deployed decision support applications are expert systems in banking area (expert systems for mortgages). The decision support systems are also widely used for river systems management to effectively cope with floods. In example The German Federal Institute of Hydrology (BfG) funded the development of a Decision Support System (DSS) for the Elbe river system. The great flooding in summer 2002 demonstrated the importance of this subject.

Some examples of DSS used in the energy sector are described in [11] [12] [13]. DSS are also successfully deployed in nuclear power plants [14], urban water pollution control [15] or oilfield flood precaution [16].

All of these DSS examples are customized and focused on some particular branch of critical infrastructures. Decision Support Systems are usually designed for special kind of industry or application. They use different methodologies (Bayesian, multi-agent, HMM), however they rarely use ontologies description to support reasoning.

INSPIRE Decision Aid Tool, presented here, is on the other hand, more general and applicable to more than one critical infrastructure. We focus rather on the used SCADA properties to enhance protection and security of critical sectors. INSPIRE Decision Aid Tool may be also considered as a framework since it is reconfigurable by means of uploading other ontologies or SCADA system topology.

4.2 Applying Ontology to INSPIRE Decision Aid-Tool

The proposed ontology mimics the complicated relationships between SCADA components and security aspects. That is why this ontology can support security solutions. However, the ontology is just a representation of relationships between particular classes (or instances) and as it is, cannot provide any knowledge-based reasoning or give feedback to its operator.

The solution is to develop the decision support tool that will map the proposed ontology into a set of rules, which will be the input for the inference engine. The engine can be described as a form of finite state machine with a cycle consisting of three action states:

- match rules
- select rules
- execute rules

During the matching process, the engine searches all rules that are matching the current content, which are also called facts. A single fact represents a single property of the real world e.g.: the operator has a SCADA network named ABC that has a vulnerability CDF that is mapped to a proper state of the engine's data store. The selected rules are the candidates to be executed, but the engine, by applying some selection strategy, determines which rule exactly will be fired. Finally, when the rule is executed, it is filled with the facts. The general DAT architecture is shown in Fig. 4.

The decision-aid analyzes the ontology and generates the facts and rules. Actually, the ontology instances are mapped to facts and the SWRL rules are mapped to the engine's rules [10]. This is achieved via the XSLT sheet that tells the parser how the ontology (OWL file) has to be transformed to CLISP language. The mapping strategy generates RDF triples from OWL file. Each RDF triple consists of:

- Predicate
- Subject
- Object

Each triple is able to fully describe one property of the instance. The interpretation of a triple is that "subject" has property "predicate" whose value is "object". Such a strategy allows DAT to be more flexible to ontology schema changes, because adding new property to the particular instance has no impact

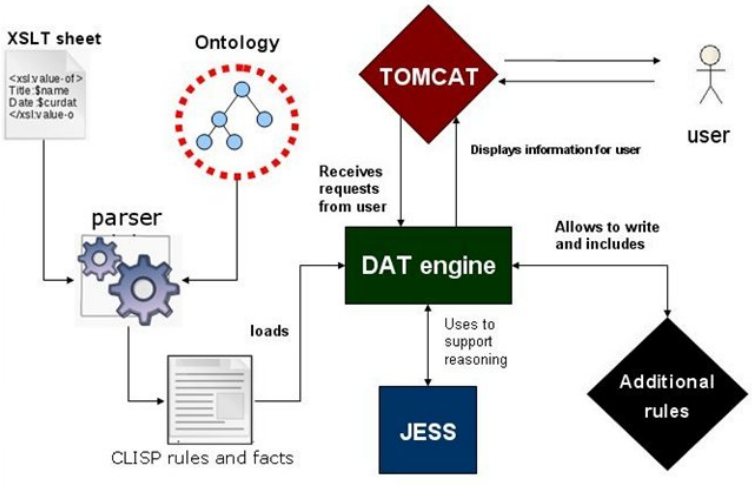


Fig. 4. DAT architecture diagram

on mapping mechanism (new triple is handled like the others one) and no impact on DAT source code. The ontology instances will be mapped to facts and the SWRL rules will be mapped to engine's rules. The SWRL fulfills the ontology functionality allowing dynamic instance and property injection. The typical SWRL rule example is often explained by means of family ontology:

"If person A has father B and person C has father B then A has brother C and C has brother A"

Thanks to such a mapping, the tool is facilitated with information that allows:

- measuring dynamically (the value will not be hard coded into ontology) the vulnerability likelihood
- dynamically ranking the vulnerabilities
- informing the operator what may cause the fault in the SCADA system
- what element of the network needs some attention and safeguards

By defining the additional (tool built-in) rules, there is a possibility to realize the customized features of the decision-aid tool:

- to propose the action plans with information how to minimize the likelihood of the vulnerability
- to apply some appropriate countermeasures to minimize the probability of the most likely attack
- to evaluate the SCADA system condition based on standards.

4.3 INSPIRE Decision Aid-Tool Description

To meet the requirements of end-users and stakeholders of Critical Infrastructures, and in order to provide reasoning, we would like to enhance the decision-aid

tool with a mechanism that would be able to answer the following questions (use cases):

1. If I have resources A,B,C, and D, what kinds of attacks should I expect?
2. If I have resources A,B,C, and D, what kinds of known vulnerabilities I might have (rank using 'risk level') ?
3. If I plan to add element A into my network, will my system become more vulnerable?
4. What kinds of safeguards should be used to eliminate vulnerabilities X, Y and Z?
5. I know my SCADA is vulnerable to attack X, are there any known safeguards?
6. Assuming my system is already vulnerable to vulnerabilities X,Y and Z, what is the probability of new vulnerabilities, threats, attacks (cascading effects)?

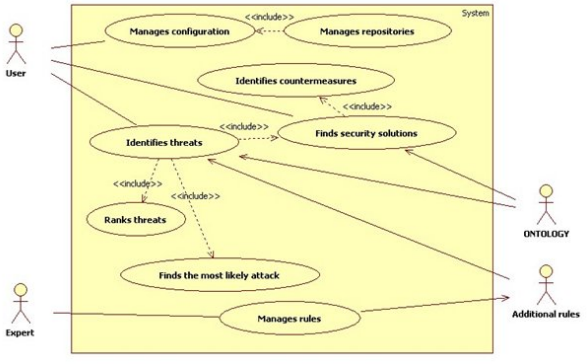


Fig. 5. DAT use cases diagram

Fig. 6. DAT welcome screen

The DAT's design describes several types of the system's actors (Fig. 5):

- User is a person that has no expert knowledge about the SCADA system and uses the DAT to identify it's system vulnerabilities and threats,
- Expert provides tool with additional customized knowledge (new rules and facts),
- Ontology Instantiator provides and manages ontology,
- Expert knowledge is provided and managed by Expert.

The welcome screen is shown in Figure 6. The INSPIRE Decision Aid Tool in operation is shown in Figures 7 and 8.

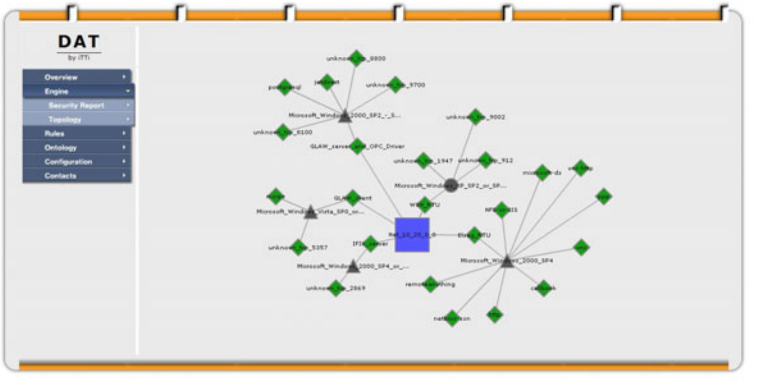


Fig. 7. DAT - example of topology visualization module

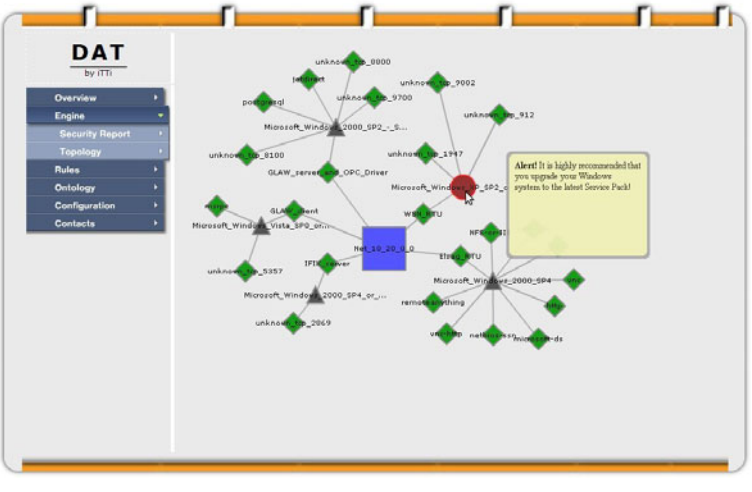


Fig. 8. Small pup-up windows allow to display significant information about discovered threat or vulnerability

5 Conclusion

In this paper an ontology-based approach for the description of SCADA systems vulnerabilities has been presented. Functionalities of the ontology applied to knowledge representation for security and protection of critical infrastructures have been shown. Our solution has been developed as a research activity in the INSPIRE Project that aims at increasing security and protection through infrastructure resilience. The presented ontology approach is the basis for the presented Decision Aid Tool (DAT) as a part of INSPIRE security framework.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 225553 (INSPIRE Project).

References

1. INSPIRE Project, Description of Work (2008)
2. D'Antonio, S., Khelil, A., Romano, L., Suri, N.: Increasing Security and Protection through Infrastructure REsilience: the INSPIRE Project. In: Setola, R., Geretshuber, S. (eds.) CRITIS 2008. LNCS, vol. 5508, pp. 109–118. Springer, Heidelberg (2009)
3. Critical infrastructure: Understanding Its Component Parts, Vulnerabilities, Operating Risks, and Interdependencies, Tyson Macaulay (August 2008)
4. Lewis, T.G.: Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation. Wiley Interscience, Hoboken (2006)
5. McClanahan, R.H.: The benefits of networked SCADA systems utilizing IP-enabled networks. In: Arkansas Electric Cooperative Corporation, IEEE, Los Alamitos (2002)
6. ISO/IEC 13335-1:2004, Information Technology - Security Techniques - Management of information and communications technology security - Part 1: Concepts and models for information and communications technology security management
7. Michał, C., Rafał, R., Adam, F., Witold, H.: Ontology-based description of networks vulnerabilities. Polish Journal of Environmental Studies, 5c (2008)
8. Stamp, J., Dillinger, J., Young, W.: Common Vulnerabilities in Critical Infrastructure Control Systems. In: Networked Systems Survivability and Assurance Department, Jennifer DePoy, Information Operations Red Team Assessments Department, Sandia National Laboratories (May 22, 2003)
9. Identification of Vulnerabilities - INSPIRE Deliverable D.2.2 (April 2009)
10. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission, http://www.w3.org/Submission/SWRL/
11. Qiong, W., Wenyin, L., Yihan, Y., Chuan, Z., Li, Y.: Intelligent Decision Support System for Power Grid Dispatching Based on Multi-Agent System. In: Proc. of International Conference on Power System Technology, pp. 1–5 (2006)
12. Xiao-Feng, D., Yu-Jiong, G., Kun, Y.: Study on Intelligent Maintenance Decision Support System Using for Power Plant Equipment. In: Proc. of the IEEE International Conference on Automation and Logistics Qingdao, China, September 2008, pp. 96–100 (2008)

13. Zhang, Z., Yang, Z., Zhang, P., Mao, Z., Hao, J.: Hierarchical Network-based Safety Assessment Decision Support System for Thermal Power Plants. In: Proc. of the 2009 IEEE International Conference on Networking, Sensing and Control, Okayama, Japan, March, 2009, pp. 592–596 (2009)
14. Lee, S.J., Mo, K., Seong, P.H.: Development of an Integrated Decision Support System to Aid the Cognitive Activities of Operators in Main Control Rooms of Nuclear Power Plants. In: Proc. of IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM), pp. 146–152 (2007)
15. Zhang, B.-P., Wu, G.-X., Shang, S.-Y.: Research on Decision Support System of Water Pollution Control Based On Immune Agent. In: Proc. of International Symposium on Computer Science and Computational Technology, ISCSCT, vol. 1, pp. 114–117 (2008)
16. Xie, L., Wang, Z., Bian, L.: The Research of Oilfield Flood Precaution Decision Support System. In: Proc. of International Seminar on Business and Information Management, ISBIM 2008, December 2008, vol. 2, pp. 236–239 (2008)
17. Michał, C., Flizikowski, A., Kozik, R., Renk, R., Hołubowicz, W.: Ontology-Based Reasoning Combined with Inference Engine for SCADA-ICT Interdependencies, Vulnerabilities and Threats Analysis. In: Pre-Proc of 4th International Workshop on Critical Information Infrastructures Security, CRITIS 2009, Bonn, Germany, Fraunhofer IAIS, pp. 203–214 (2009)

Application Filters for TCP/IP Industrial Automation Protocols

Aguinaldo B. Batista Jr., Tiago H. Kobayashi, João Paulo S. Medeiros, Agostinho M. Brito Jr., and Paulo S. Motta Pires

LabSIN - Security Information Laboratory
Department of Computer Engineering and Automation
Federal University of Rio Grande do Norte
Natal, RN, Brazil
{aguinaldo,hiroshi,joaomedeiros,ambj,pmotta}@dca.ufrn.br

Abstract. The use of firewalls is a common approach usually meant to secure Automation Technology (AT) from Information Technology (TI) networks. This work proposes a filtering system for TCP/IP-based automation networks in which only certain kind of industrial traffic is permitted. All network traffic which does not conform with a proper industrial protocol pattern or with specific rules for its actions is supposed to be abnormal and must be blocked. As a case study, we developed a seventh layer firewall application with the ability of blocking spurious traffic, using an IP packet queueing engine and a regular expression library.

Keywords: Industrial Firewall, Critical Information Infrastructure Protection, Modbus/TCP Protocol.

1 Introduction

Industrial communication systems are characterized by the use of a wide variety of restricted communication technologies with different features and purposes. However, it is notorious that some characteristics of industrial communications systems have been in change since the last decade. There is an undoubted trend towards the vertical integration between the various layers of the Automation Technology (AT) pyramid, which leads to an increase in connectivity, flexibility and productivity of these systems, followed by a substantial cost reduction. This integration have been facilitated by the progressive substitution of specific, proprietary and closed technologies utilized in these systems, for open, standardized and general purpose ones. Among these technologies are the TCP/IP-based networks whose increasingly deployment in industrial systems constitutes a factual tendency.

There are many industrial protocols running over a TCP/IP basis. Some of them are newer ones designed especially to run over TCP/IP while others are slightly adapted versions of well established industrial protocols which are encapsulated as application layer protocols in the TCP/IP reference model. Consequently, in a recent past, information security issues related to the utilization of

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 111–123, 2010.

Information Technologies (IT) in automation systems has become an important concern. The public exposure of some security incidents [1,2] has attracted the attention of AT community specialists and regulation institutions.

Security of automation systems has become a common topic of interest discussed in several works over the last years. The subjects range from the automation security assessment to countermeasure proposals for securing automation systems [1,2,3,4,5,6,7,8,9,10,11]. Some of these proposals advise the utilization of different types, classes and architectures of network firewalls to protect automation systems from malicious activity [8,9]. Given that firewalls are filters by nature and they are widely used in the IT sector, the firewall approach is strongly recommended to secure AT networks from IT networks threats. However, this may not be enough as automation systems have relied in TCP/IP networks even in lower layers of the automation pyramid and therefore have become more susceptible to remote and insiders' (maybe not intentionally realized) attacks. Hence, the firewall approach can go deeper and may be also suitable to secure lower layers of the automation networks, where time is critical and devices perform delicate tasks. It is supposed that firewalls applied to such environments should be able to understand automation protocols, accomplishing a deep inspection traffic filtering.

The objective of this work is to propose a comprehensive industrial protocol filtering system for TCP/IP automation networks in which malicious activities can be minimized by blocking all non-industrial traffic (traffic which does not match a determined protocol pattern) and industrial traffic related to non-permitted actions. The firewall system in our proposition has as main component an user-space application which analyses application layer traffic. This application basically uses an IP packet queueing facility for traffic processing in user-space and a regular expression library for pattern matching. Modbus/TCP was the automation protocol chosen as a case study but this approach can be extended to other protocols running over TCP/IP.

The remainder of this paper is organized as follows. Section 2 presents the paper background which is founded in network firewalls and in the Modbus/TCP protocol. Section 3 introduces the reader to the problems that have motivated this work. The proposed solution is briefly explained in Section 4. Section 5 lists and describes the tools used in the development of our firewall system while Section 6 details the system architecture. We show some results in Section 7 and we finaly present some final considerations and proposals for future works in Section 8.

2 Paper Background

2.1 Network Firewalls

Network firewalls are information security devices which monitor and regulate traffic on a computer network. They can protect information, services and devices on a network from series of threats like attacks, non-authorized access or resource misuse. Firewalls are commonly used to segregate networks from

different reliability levels. For example, in the automation systems information security context, firewalls have been strongly recommended as an effective way to protect industrial TCP/IP networks from corporate networks.

Firewalls act by inspecting traffic accordingly to rules defined for several layers (numbered from 1 to 7) of the Open System Interconnection (OSI) reference model. Conventional firewalls, also known as packet filters, inspect network traffic under the layers 3 and 4 (network and transport layers, respectively). That is, they can do basically rule matching over IP addresses (layer 3) and port numbers (layer 4). Although packet filters are enough for most situations, they are not able to do rule matching over data situated in the application layer because this kind of firewall simply cannot understand protocols placed in the application layer (the payload of the transport layer, in the case of the TCP/IP networks) which is the case of automation protocols based on TCP/IP. It is needed to endow this firewalls with the ability of inspecting the 7th layer, so it will be possible to realize the regulation of automation protocols' traffic. Firewalls with this aptitude are known in the specialized literature as application or layer-7 firewalls.

This approach for firewalls is already widely utilized em some IT environments, especially in situations where the conventional filtering is not enough to grant the desired level of security or save network bandwidth. An example of a well succeeded application firewall for IT environments is the open-source *l7-filter* [12]. It is a classifier module for the Linux's Netfilter [13] that identifies connections' application layer data. It was designed to match some common IT application protocols packets, specially chatting and P2P (peer-to-peer) ones. The *l7-filter* utilizes regular expression libraries to determine what protocol is being used. It does not run a regular expression matcher on every upcoming packet. Instead, to avoid the wasting of clock cycles, *l7-filter* just looks at the first few packets of a connection for protocol-specific messages which are defined in its protocol pattern database.

As TCP/IP networks have arrived to the industrial environment and some traditional industrial protocols have been ported to these networks as application protocols, it is not surprising that people of the AT sector had started to think about developing application filters for these protocols. Cisco's CIAG (Critical Infrastructure Assurance Group) launched in 2004 a industrial networking research project which produced an open source industrial protocol-aware firewall solution which consisted of a Linux's Netfilter extension capable to inspect Modbus/TCP packets. This extension, called Modbus Firewall [14], was able to match Modbus/TCP packet fields values with some user defined values in order to block undesirable Modbus/TCP packets. The solution was designed to evaluate the feasibility of adding fine-grained access controls for an automation protocol within a general purpose firewall environment. The main drawback of this application firewall is that it is quite limited as it lacks some important matching features and it concludes Modbus/TCP packet field matching without verifying if the protocol data under analysis is really Modbus/TCP traffic.

2.2 Modbus/TCP Protocol

Modbus [15] is one of the oldest and most used protocols designed for automation systems. The original Modbus specification ran on RS-232, but most later Modbus implementations used RS-485 [15,16].

Modbus/TCP protocol embeds a standard Modbus serial data frame, formally called Application Data Unit (ADU), into the payload of a TCP segment, excluding the Modbus Checksum and Address fields, and adding an specific header to the remaining fields, as shown in Figure 1. The Modbus Application Protocol (MBAP) Header is a 7-byte structure that includes the following fields [17,18]:

- **Transaction Identifier (2 bytes):** It is an identification field used for transaction pairing. It distinguishes Modbus/TCP transactions when multiple messages are sent along the same TCP connection by a client.
- **Protocol Identifier (2 bytes):** This field identifies Modbus/TCP and it is used for intra-system multiplexing. Its default value is 0.
- **Length (2 bytes):** The value of this field represents a byte count of the remaining fields of the packet. It contains the sum of the unit identifier, function code and data field lengths.
- **Unit Identifier (1 byte):** This field is used for intra-system routing purposes. It identifies a remote server located on a non-TCP/IP network, when using a Modbus/TCP to Modbus serial bridging scenario.

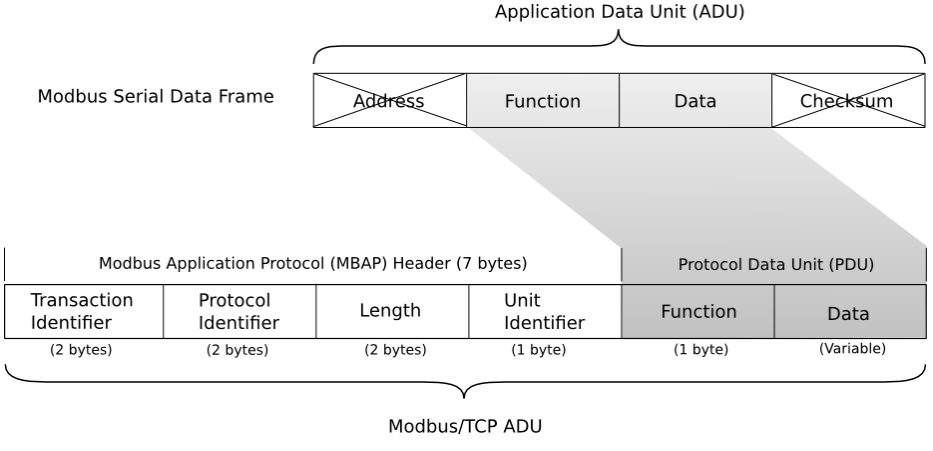


Fig. 1. Construction of a standard Modbus/TCP data packet

As illustrated in Figure 1, the building process of a Modbus/TCP packet strips down a traditional Modbus RTU serial data frame, by excluding the Address field (Address), which is supplanted by the Unit Identifier field on MBAP Header and the error detection field (Checksum), once Modbus/TCP relies on the standard Ethernet and TCP/IP layer checksum methods instead. Modbus Function Codes with the Data field, which together compose the Protocol Data

Unit (PDU), are concatenated with the MBAP Header, generating the Modbus/TCP ADU data packet. This data is encapsulated into the payload of a TCP segment in a established TCP connection and transmitted over a IP-based network. The Figure 1 also shows the standard Modbus/TCP packet format, which can carry three types of Modbus/TCP messages: request, response and exception. Request messages are sent by the client to the server while the server answer these requests with response or exception messages, without modifying the standard packet format.

3 Problem Definition

Our disposal to develop a specialized filter can be justified by the results of studies and tests developed in our laboratory about the security of automation network, protocols and devices. As a result, two scenarios where the presence of undesirable traffic could comprise risks to the integrity of automations networks and devices came up. The first is the presence of non-industrial traffic in automation networks and the second is the misuse of protocol's actions by malicious entities.

3.1 Non-industrial Traffic in Automation Networks

The presence of non-industrial traffic in automation networks may represent a serious risk to the security and integrity of industrial networks. Critical information could be improperly accessed, network scanners could be used to discover network inventory and topology, undesirable latency could be insert, and denial-of-service (DoS) attacks could be done to saturate network bandwidth or harm automation devices.

The harmful effects of non-industrial traffic in automation networks could be proved in practice [19]. The problems may not be related to vulnerabilities of the Modbus/TCP protocol itself, but they could be provided by an implementation that do not conform to the protocol specification or a problem in the device's TCP/IP protocol stack. It comes up the need for a filtering device which blocks all non-industrial traffic in an automation network, that is, it must permit only traffic which complies with a determined protocol pattern. Once this kind of filtering is utilized, the traffic in the protected segments would be assuredly known and the worrying about the mentioned problems could be minimized.

3.2 Misuse of Protocol Actions

A practical scene of misuse of protocol actions can be visualized in a work about SCADA malwares [20]. In this work, authors present a scenario where automation plant is infected by malwares specifically designed for this kind of environments. It is utilized a mobile-agent platform which simulates the behaviour (infection and propagation) of several families of malwares. These malwares are specially programmed to affect Modbus/TCP automation devices and they are

capable of launching two types of attacks: denial-of-service attacks and protocol misuse attacks. The former aims to desynchronize clients and servers by bandwidth saturation or by consuming all devices' resources. Fortunately, for this kind of attack, most modern firewall can deal with the problem by rate limiting or flood blocking. The latter aims to explore protocol actions in a malicious way. Among the protocol actions performed are unauthorized write operations on a large set of registers a time and the inversion of multiple register state. These actions, like many others, constitute legitimate Modbus/TCP actions but the they might be harmful to the pant in determined situations.

This kind of scenario may justify the need for a traffic blocking system for denying clients on performing some actions, accordingly with a previous knowledge about characteristics and limitations of plant devices.

4 Proposed Solution

Our proposed solution consists in an application firewall system which is able to block all non-industrial and undesirable Modbus/TCP traffic. This system has as the main element an user-space application which inspects network traffic in the application layer. It uses a packet queueing mechanism that permits packets to be processed in user-space. This packet processing procedure consists on the analysis of the payload of TCP segments.

As the situations and problems occurred during the study of Modbus/TCP devices, this protocol was chosen as a case study. Nevertheless, this proposed approach can be applied to any TCP/IP based automation protocol.

The analysis process is divided in two sequential phases. In the first phase, conformance tests based on regular expressions matches are executed to the payload of TCP segments, to assure that non-industrial traffic will not populate de industrial network. After this conformity test, client-side traffic (Modbus/TCP requests) is submitted to the second phase which consists on the analysis of protocol fields against user-defined rules for each Modbus/TCP devices in the network. The processing done in this phase assures that no malicious, harmful or out-of-context requesting actions will get to the protected devices.

5 Tools and Techniques Used

5.1 Traffic Regulation on Linux

Traffic regulation in the Linux OS is usually achieved by the Netfilter/Iptables couple. Netfilter is a Linux kernel subsystem that handles TCP/IP traffic while Iptables is the user space application which converses with Netfilter through matching rules defined by the user. It is possible to define Iptables rules that will match packets and decide the actions that will be taken.

The main advantage of building a specific firewall system based on this infrastructure is the use of Iptables as a complementary tool for supplying the absences

of the specific-purpose firewall. Thus, Iptables rules can be normally used for layer 3 and 4 matching purposes.

5.2 Packet Queueing

When developing Linux-based firewalls, the most common approach is to develop series of Netfilter (kernel-space) and Iptables (user-space) patches, which is not a simple task as it demands much time (for kernel recompiling and testing) and requires a reasonable knowledge on Netfilter/Iptables functioning, C language programming and kernel patching. The alternative to this approach is to use a packet filtering scheme where packets would be caught in the kernel space, brought to the user space where they would be analyzed and then allowed or disallowed to continue its path through the network.

The existing way of doing that is to use the Netfilter/Iptables engine for user space packet queuing [21]. This engine has the ability of passing IP packets out of the stack for queueing to user space, then receiving these packets back into the kernel with a verdict specifying what to do with them (actions such as *ACCEPT* or *DROP* the packets). These packets may also be modified in user space prior to re-injection back into the kernel. For each supported protocol, a kernel module, called queue handler, may register with Netfilter to perform the mechanics of passing packets to and from the user space [22].

The standard Netfilter queue handler for IPv4 is *ip_queue* and it is provided as an optional stable module with Linux 2.6 kernels. This module employs a Netlink socket as the communication channel between user-space and kernel-space [23]. Once *ip_queue* module is loaded a new action, also known as target or jump, called *QUEUE*, is added to the possible actions. So, IP packets can be matched with Iptables filtering rules (using the filter table) and queued for user space processing through the now available *QUEUE* target. A typical Iptables filtering rule specifies the chain where the packet will be filtered (*INPUT*, *OUTPUT*, *FORWARD*), several matching parameters like the affected protocol, source/destination IP/ports or network interfaces and, finally, an action that defines what to do with a packet. For example, running the following command:

```
# iptables -A INPUT -p tcp -j QUEUE
```

an Iptables rule is added (*-A*) establishing that all TCP segments (*-p tcp*) that get to the *INPUT* chain (where filtering is done for all incoming traffic destined to our localhost) will be submitted (*-j*) to the *QUEUE* target (action). This action will queue incoming packets to user space applications. Once the packet is available in the user space, we can access the packet data by using the user space packet queuing library, *Libipq* [22], which provides an Application Programming Interface (API) for communication with the *ip_queue* handler. Thus, any application using *Libipq* may receive the queued packets and will able to process them in user space practically the same way as in the kernel space. Hence, it is possible to do almost everything with packets, like set verdicts (*ACCEPT* or *DROP* the packets), mark them or even manipulate packet's content.

The main advantages of this approach is that there is no need of Linux network internals expertise. A programmer would focus attention on writing his own packet mangling or firewall applications in the user domain, without having to worry about kernel coding.

5.3 Regular Expressions

Regular expressions are a notation for describing a certain set of character strings. When a particular string belongs to the set described by a regular expression, it is said that the regular expression matches the string.

There are many ways to get closer to regular expressions matching. One of the most powerful and popular regular expression library, available by default or easily installable in most Linux distributions, is the Perl Compatible Regular Expressions (PCRE) [24] library, used in this work to match our case study protocol. This library was chosen due to its simplicity, popularity and extensive documentation which is complemented with Perl regular expressions documentation [25].

Our user-space application makes use of the PCRE library as the main validating system that matches the Modbus/TCP protocol pattern. Captured traffic is submitted to a conformity analysis, using regular expressions developed to match the Modbus/TCP protocol traffic.

5.4 Inventory-Based Rules

After the regular expression matching stage, Modbus/TCP approved traffic is submitted to the second processing phase, where protocol fields are checked against user-defined rules. These rules are defined in a device inventory fashion, in which devices in the automation network are defined in a rule file. In this file, Modbus/TCP devices are specified in terms of their constraints for all Modbus/TCP PDU fields. Hence, network administrators can define the permitted actions for each Modbus/TCP device individually in an easy and straightforward way.

6 System Architecture

Our application firewall is supposed to run on a Personal Computer (PC) hardware platform, with at least two network interface cards (Ethernet) and running a Linux (kernel 2.6) operating system. The Linux system must have its packet routing capabilities activated because it will act as a router between the client-side and the server-side networks. An Iptables rule must be defined in this system to forward the interest traffic (IP packets with TCP segments) to the QUEUE target (available after loading the *ip_queue* module). The following command established this Iptables rule:

```
# iptables -A FORWARD -p tcp --sport 502 --dport 502 -j QUEUE
```

This rule forwards packets containing TCP segments from both client and server sides to a queue whose head is accessible in the user-space.

This system runs the firewall application itself, named Modbus filter. This application retrieves packets from the head of packet queue and inspect them, deciding, after that, if packets will be accepted or dropped. Figure 2 illustrates the system architecture.

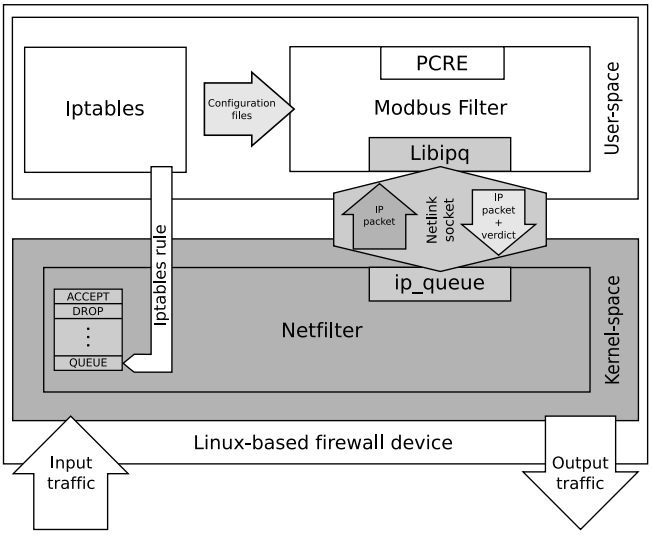


Fig. 2. System Architecture

The developed application is written in C language. It retrieves IP packets from the queue and submits the payload of the TCP segment to series of tests that will determine if the corresponding IP packet will be accepted or dropped. The library `libipq` is used to retrieve packets and the library PCRE is used to match application layer data with Modbus/TCP regular expressions. The application reads a configuration file which specifies things like the IP address of the firewall interfaces, the regular expressions and some options. A rule file is also read to let the application know the devices inventory and constraints.

TCP control segments are freely allowed to pass through the firewall. Non-control segments from both sides (client and server) are analyzed in terms of the Modbus/TCP regular expressions while only segments from the client side (Modbus/TCP requests) are submitted to the rule conformity analysis. Various complementary validation checking are also made. One of them verifies if the length of the TCP payload attends to the maximum Modbus/TCP ADU length while another checks if the length indicated by the Length field in the MBAP header constitutes the real value.

7 Tests and Results

7.1 Testbed

The established testbed reproduces, in a simplified way, a Modbus/TCP industrial network with clients and servers. This testbed is shown in Figure 3. The testbed is composed of two networks (client-side and server-side networks) interconnected through a dual-homed PC which runs our application firewall on a router-enabled Linux OS. Both client-side and server-side networks are monitored with the help of Test Access Point (TAP) devices and monitor PCs. Client-side machines are PCs acting as Modbus/TCP clients by running a Modbus/TCP packet manipulator software that can inject any kind of Modbus/TCP traffic as well as malformed Modbus/TCP traffic [19]. This clients can also inject various types of IT traffic, since ordinary traffic (common IT traffic from various network applications) to malware and attack traffic. Server-side machines are PCs that plays the role of Modbus/TCP server devices by running a Java implementation of the Modbus/TCP protocol, called *jamod* [26]. This implementation can behave as a Modbus/TCP server that can listen to the 502/TCP port and respond to Modbus/TCP requests.

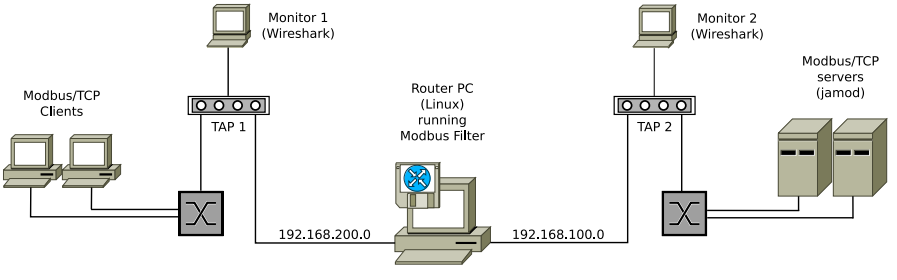


Fig. 3. Testbed established for firewall testing

7.2 Tests Realized

It was established a test schedule involving the injection of various types of traffic into the server-side network, to verify the functioning and effectiveness of the developed firewall system. Typical IT network traffic from network scanners (nmap, nessus), remote access tools (telnet, rsh, ssh) and common IT malwares was summarily blocked by the firewall. Except for TCP control segments, all traffic from these sources is blocked because it did not match the regular expression or did not pass the complementary validation checking. Modbus/TCP requests that did not conform with the permitted actions described in the rule file were also successfully blocked.

It is important to highlight that the discarding of IP packets containing TCP segments carrying non-conformity data interferes in the TCP stream of a

Modbus/TCP transaction. As no component part of the connection has any knowledge about this packet drop, series of events are launched in both sides of the connection. The behaviour of clients and servers on packet loss depends on the implementation of the software which runs on devices (TCP/IP stack and Modbus/TCP protocol implementation) and must be analyzed before deploying this kind of firewall system.

7.3 Performance Results

Firewall performance can be measured under several metrics, according to its sophistication degree. The chosen metric was the latency increase introduced by our firewall system on the mounted testbed. This choice was taken because latency is an extreme importance parameter in critical systems and should not be overlooked.

To measure latency increase introduced by our firewall, it was used a technique commonly used to avail the behaviour of TCP/IP networks. This technique consists on the measurement of the round-trip time (RTT) of information. That is the time interval taken by an information to get to the receiver and its acknowledge or response to get back to the sender. In the case of the TCP protocol, this interval is computed as the difference between the time taken by a segment to be sent and its confirmation (ACK segment) to be received.

This performance study covered our firewall system as whole device, that is, it covered the latency increase introduced by all hardware and software components of the system. The analysis of the RTT samples under several situations showed that our firewall system introduces, in average, a latency increase.

8 Conclusion and Future Works

This work showed how an application filter for TCP/IP industrial networks can be developed in a easy and straight-forward way. An application written in the user-space can make use of the Linux Netfilter/Iptables infrastructure and regular expression libraries to analyze and regulate traffic in a industrial network, impeding malicious traffic from harming a critical infra-structure. However, the firewall system introduces a considerable latency increase and interferes in the TCP stream. These effects should be carefully analyzed before the deployment of such firewall in a real world critical systems.

Future works may include the extension of this approach to other TCP/IP industrial protocols, like DNP3/IP, Ethernet/IP and IEC 60870-5-104 and the refining of regular expressions of each protocol. Another perspective comprises the utilization traffic analyzed by the firewall as a basis for an intelligent anomaly detection system in industrial networks. From previous observations on validated traffic, an intelligent system could detect anomalies in the standard behaviour of plant devices. Based on detections, alarms could be triggered and new firewall rules could be dynamically created.

Acknowledgments. The authors would like to express their gratitude to the Department of Computer Engineering and Automation from Federal University of Rio Grande do Norte, Brazil, and REDIC (Instrumentation and Control Research Network) for the support received along the development of this work.

References

1. Byres, E., Hoffmann, D.: The Myths and Facts behind Cyber Security Risks for Industrial Control Systems. Technical report (2003)
2. Creery, A., Byres, E.: Industrial Cybersecurity For Power System And Scada Networks. In: 52nd Industry Applications Society Conference on Petroleum and Chemical Industry, pp. 303–309 (2005)
3. Pires, P., Oliveira, L.: Security Aspects of SCADA and Corporate Network Interconnection: An Overview. In: Proceedings of International Conference on Dependability of Computer Systems, DepCoS-RELCOMEX, Szklarska Poreba, Poland, pp. 127–132 (2006)
4. Krutz, R.L.: Securing SCADA Systems. Willey, Indianapolis (2006)
5. Treytl, A., Sauter, T., Schwaiger, C.: Security Measures for Industrial Fieldbus Systems - State of the Art and Solutions for IP-based Approaches. In: Proceedings of IEEE International Workshop on Factory Communication Systems, September 2004, pp. 201–209 (2004)
6. Dzung, D., Naedele, M., Hoff, T.P.V., Crevatin, M.: Security for Industrial Communication Systems. Proceedings of IEEE 93, 1152–1177 (2005)
7. Byres, E., Karsch, J., Carter, J.: NISCC Good Practice Guide on Firewall Deployment for SCADA and Process Control Networks (February 2005)
8. P. C. Group, NISCC: Good Practice Guide: Process Control and SCADA Security (October 2005)
9. Stouffer, K., Falco, J., Kent, K.: Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security. NIST Special Publication (800-82) (September 2006)
10. Paukatong, T.: SCADA Security: A New Concerning Issue of an In-house EGAT-SCADA. In: 2005 IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific, pp. 1–5 (2005)
11. Pollet, J.: Developing a Solid SCADA Security Strategy. In: Sensors for Industry Conference (Sicon/02), pp. 19–21 (2002)
12. l7 filter: Application Layer Packet Classifier for Linux (2009), http://l7-filter.sourceforge.net
13. Netfilter.org: Linux Netfilter (2009), http://www.netfilter.org
14. Franz, M., Pothamsetty, V.: Transparent Modbus/TCP Filtering with Linux (2004), http://modbusfw.sourceforge.net/
15. Modbus-IDA: Modbus Application Protocol Specification. Modbus-IDA (December 2006)
16. Bies, L.: Modbus Interface Tutorial. Technical report (2009)
17. Acromag: Introduction To Modbus TCP/IP. Acromag Incorporated (2005)
18. Modbus-IDA: Modbus Messaging on TCP/IP Implementation Guide. Modbus-IDA (October 2006)
19. Kobayashi, T.H., Batista, A.B., Brito, A.M., Motta Pires, P.S.: Using a Packet Manipulation Tool for Security Analysis of Industrial Network Protocols. In: IEEE Conference on Emerging Technologies and Factory Automation, pp. 744–747. ETFA (September 2007)

20. Carcano, A., Fovino, I.N., Masera, M., Trombetta, A.: Scada Malware, a proof of Concept. In: Setola, R., Geretshuber, S. (eds.) CRITIS 2008. LNCS, vol. 5508, pp. 247–257. Springer, Heidelberg (2009)
21. Netfilter: Linux Netfilter Hacking HOWTO (2009), http://www.netfilter.org/documentation/HOWTO/netfilter-hacking-HOWTO-4.html
22. Libipq: Libipq - Iptables userspace packet queuing library (2009), http://linux.die.net/man/3/libipq
23. Benvenuti, C.: Understanding Linux Network Internals. O'Reilly, Sebastopol (2005)
24. PCRE: Pcre - perl compatible regular expressions (2009), http://www.pcre.org
25. Perl: perlre - perl regular expressions (2009), http://perldoc.perl.org/perlre.html
26. Jamod: jamod (2009), http://jamod.sourceforge.net

Web Browser Security Update Effectiveness

Thomas Duebendorfer[1] and Stefan Frei[2]

[1] Google Switzerland GmbH
[2] Swiss Federal Institute of Technology (ETH Zurich)
insecurity-iceberg@tik.ee.ethz.ch
http://www.techzoom.net/silent-updates

Abstract. We analyze the effectiveness of different Web browser update mechanisms on various operating systems; from Google Chrome's silent update mechanism to Opera's update requiring a full re-installation. We use anonymized logs from Google's world wide distributed Web servers. An analysis of the logged HTTP user-agent strings that Web browsers report when requesting any Web page is used to measure the daily browser version shares in active use. To the best of our knowledge, this is the first global scale measurement of Web browser update effectiveness comparing four different Web browser update strategies including Google Chrome. Our measurements prove that silent updates and little dependency on the underlying operating system are most effective to get users of Web browsers to surf the Web with the latest browser version.

1 Introduction

Our global scale measurements of Web browsers in use [1] from mid 2008 found that 45% of Internet users were not using the latest Web browser version when visiting Google Web servers. If people keep using an outdated Web browser version with known vulnerabilities, they can easily fall victim to any of the millions of malicious Websites that execute drive-by downloads to infect the visitor's computer with malware. In April 2009, Finjan discovered a bot network of more than 1.9 million [3] infected computers, which was built up since February 2009 by using drive-by downloads as primary infection channel.

In June 2008, we found the Mozilla Firefox Web browser to have the most effective update mechanism of any popular browser [4]. However, throughout June 2008, at most 83% [1] of all active Mozilla Firefox users were using the latest version. We were wondering if one cannot do even better than Mozilla Firefox by deploying a different update mechanism in a Web browser.

The Google Chrome Web browser [5], which was released to the public as beta in September 2008 and came out of beta in December 2008, is using a so-called silent update mechanism. The user currently cannot disable auto-updates in Google Chrome, which is different from any other browser update mechanism in use today. This gave us a great opportunity to evaluate the effectiveness of Google Chrome's update mechanism by comparing it to other Web browsers using the same data source and a similar measurement methodology as used in our two previous Web browser studies [1,4].

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 124–137, 2010.

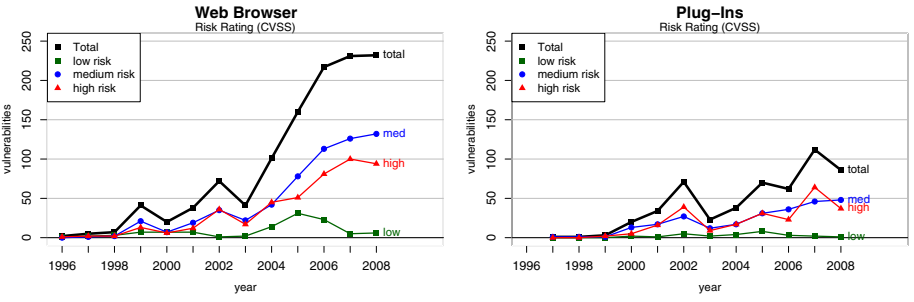


Fig. 1. Number of new vulnerabilities reported yearly for Web browsers and popular plug-ins split by CVSS risk rating. Source: National Vulnerability Database (NVD).

2 Why Update Effectiveness Matters

In recent years, major software vendors have considerably increased the share of patches availabe on the day of vulnerability disclosure [2]. However, these efforts are in vain if the user does not actually install the patch.

Keeping software up-to-date brings several benefits to the user:

- increased security thanks to timely deployment of security vulnerability fixes
- better software stability thanks to timely bug fixes
- new features that make software more powerful

At the same time, getting all users to work with the latest software release is also advantageous for the vendor:

- most likely happier users due to more stable, more secure applications with additional features
- less support required: only unfixed bugs in the latest version get reported by users
- less testing: engineers don't have to keep testing older versions on newer platforms and with new third party software or drivers

Now, imagine the user experience of your software, if a third or more of the user base is not using the latest version as this was the case for Apple Safari, Opera, and Microsoft Internet Explorer in mid 2008. This large user base will never see the improvements and new features of the latest version - and will be unnecessarily exposed to old threats.

In the optimal case, all users would always be using only the latest release of a software. However, this also introduces a big drawback: The software gets even more dynamic as each time it is used, it could be a different version with some unexpected features, new behavior and possibly new bugs. This takes control away from the user and gives it to the software vendor, which some power-users don't like. Furthermore, some developers need to control which version of a software they run, mostly for compatibility testing purposes. However, most

of this testing would no longer be needed if all users were always using the latest version.

Interestingly, this optimal case is actually widely in use nowadays with software as a service in the form of Web based (e.g. AJAX) applications. Code in many Web application implementations changes frequently without the user even noticing it as most is done "under the hood" and not visible in the user interface. The large majority of Web applications does not even expose a version number to the end user because there's simply no need to care about updates of Web applications for the user. The underlying Web servers and frameworks often expose a version number in the HTTP headers. However, we're speaking here of the actual business logic code built on top of it. Despite the "loss of control" argument, most users have accepted this fact. We think that using silent updates makes sense also for many programs installed on end user systems. In the end, it should not matter for the user where and how the application is installed as long as a high service quality, compatibility and appropriate user data privacy is respected.

3 Trends in Web Browser and Web Server Insecurity

Tracking the number of vulnerabilities discovered each year in Web browsers from year 1996 to 2008 as shown in Figure 1 reveals a clear trend: The total number of Web browser vulnerabilities reported in the National Vulnerability Database (NVD) [7], which were rated low, medium or high risk in the Common Vulnerability Scoring System (CVSS) [8], was increasing rapidly year over year since 2003. From 2000 to 2007, the number of new vulnerabilities discovered per year in popular Web browser plug-ins has more than quadrupled. From 1996 to 2008, the most vulnerable plug-ins were Apple Quicktime with a staggering number of 125 vulnerabilities found and Adobe Acrobat with 59 vulnerabilities reported.

On the other side, classical Web servers (i.e. Apache HTTP Server, Microsoft Internet Information Server, Netscape Enterprise Server, IBM HTTP Server and Sun One Web Server) had their last peak of newly reported vulnerabilities in 2002 as shown in Figure 2. While classical Web servers became less vulnerable over time, new Web application servers and scripting engines became popular and introduced new vulnerabilities in Web applications. For the PHP scripting engine, 116 vulnerabilities were reported in year 2007 alone, without counting vulnerabilities found in PHP based Web applications (like php-nuke, phpbb, Typo3 and many more). Attackers exploit such vulnerabilities in Web applications and misuse them to initiate drive-by downloads of malware to infect visiting vulnerable Web browsers.

Combined with the fact that most vulnerabilities were exploitable from the network and almost none needed local access, the Web browser has clearly become a weak link in the chain of systems that make the Internet work. Extrapolating from the past, it's very likely that Web browsers will have vulnerabilities rated medium and high risk and which are exploitable from remote in the near future as well. A good practice to keep the network and end-users secure is to

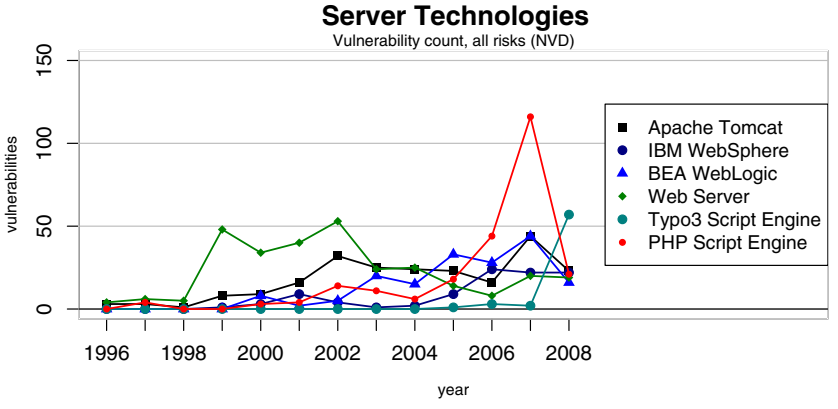


Fig. 2. Number of new vulnerabilities reported yearly from 1996 to 2008 for Web servers and various Web application servers. Source: National Vulnerability Database (NVD).

patch known vulnerabilities as fast as possible. However, to do so, Web browsers are in dire need of a very effective update mechanism or they will lose the battle for securing vulnerable Web browsers before their users fall victim to attackers.

4 Update Mechanism Variety

We consider the different types of update mechanisms in use by software today. The core steps on the client side in an update process are:

- discovery: learn about availability of the update
- download: copy the update files to the local system (after checking authenticity of the update source)
- installation: modify local installation with changes introduced by the update (after checking integrity of the update)
- activiation: execute the newly installed software to let changes take effect

While each step looks trivial, the underlying question to ask is how the delay from update availability to when the update is applied by end users is influenced in each step of any specific update mechanism in use. This ultimately reflects in the effectiveness of the update mechanism as measured by the share of active users that have updated their software after the release of a new patch. There are other factors as well such as update policies (i.e. compatibility requirements with existing (e.g. intranet) applications, or a company wide schedule for updates etc.), willingness to update (i.e. a new user interface might let people block an update) and possibly licence cost (not applicable for most Web browsers as they are available for free). We'll briefly summarize how the five most popular Web browsers get updated in Tab. 1. Each Web browser software listed here also allows to manually check for updates. All of the described systems poll for

updates from update servers and none of them gets updates pushed directly. This is mainly for security reasons, which often prevent Internet servers from contacting clients directly. However, for the user, this does not make a difference besides an additional delay introduced for discovering a new update. Given that each browser considered here uses a different update system and strategy, it's clear that industry has not yet settled on a single best practice yet. The only common thing we found is that any installed update will not be applied until a restart of the browser. A more detailed description of each browser's update mechanism and settings can be found in our tech report [6].

Table 1. Comparison of Web browser update mechanisms

Web browser	**discovery interval**	**download mechanism**	**installation procedure**
Google Chrome	every 5 hours	automatically	automatically; browser keeps running during installation; user is not prompted to restart when new version is installed
Mozilla Firefox	once per session, each-time, when-appropriate, or never	automatically, manually, or never	manually with one mouse click; browser must be closed during installation and will restart afterwards
Apple Safari	daily, weekly, monthly, or never	manually, optionally automatically for important updates, or never	manually (often together with other updates); browser sometimes must be closed during installation
Opera	weekly, or never	manually or never	manually (like a fresh browser install from scratch); browser must be closed during installation
Microsoft Internet Explorer	daily, weekly, or never	automatically, manually. or never	automatically or manually; browser sometimes must be closed during installation

After publication of our browser study [1] in July 2008, we got many write-ins of users explaining to us why they prefer not to update their Web browser. Some users simply don't want to update because updating can be very inconvenient. Some update mechanisms interrupt the user at work, require waiting during the download, prompt for installation instructions, and block the user during the actual installation of the update. Finally some updates break the system or expose the user to a new user interface, which requires time to get used to or which the user simply does not like better than the old one. In addition, there's no guarantee that an update can be undone without side effects. Especially with larger updates, the benefits of installing a new version sometimes do not outweigh all the troubles an update could cause. If update mechanisms were designed with ease of use and convenience in mind, users would be much more willing to get updates deployed.

In terms of user interaction, only Google Chrome's silent update mechanism does not disturb the user at all. The user is never disturbed working by getting prompted for download or installation of an update. The update will simply be applied the next time the user decides to restart the browser. While this totally silent update causes no disruption to the user, it has the disadvantage that the user is not actually made aware of the update and might keep running an outdated version for days or weeks, even though a newer version is already installed on the local system. At least for security bugs, it would seem advantageous to make the user aware of the installed update and ask for a browser restart at the next convenience.

5 Measuring Update Effectiveness

To measure update effectiveness, we looked at the percentage of daily active users that use the newly released Web browser version; with 100% being all users of the same major version seen on the same day. By tracking the usage shares over three weeks after a new release, we could determine how fast users update to the latest version and compare the update performance between different releases of the same and other browsers.

We used anonymized logs from Google's world wide distributed Web servers and parsed the HTTP user agent string, which each browser sends to any Web server, when requesting a Web page. The user agent string contains the browser's name and version number as well as some other information. Here are two sample user agent strings of Mozilla Firefox 3.0.8 and Google Chrome 1.0.154.53:
1) `Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.0.8) Gecko/2009032609` **`Firefox/3.0.8`**
2) `Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.19 (KHTML, like Gecko)` **`Chrome/1.0.154.53`** `Safari/525.19`

For eliminating duplicate visits on the same day, we counted only the first Web request by each Web browser with the same Google PREF cookie. We used the Pacific Daylight (Savings) Time timezone as day reference because all but one vendor of the considered Web browsers have their head quarters in this time zone (with the exception of Opera, which is based in Norway). We split the version number in major and minor versions, e.g. Mozilla Firefox 3.0.8 has a major version of 3. Upgrading between major versions is a much bigger hurdle for users as the changes in the software versions are larger and sometimes introduce incompatibilities and user interface changes. For this paper, we focused on the update effectiveness within the same major version of various Web browsers. We did not consider "upgrades" between different major versions in our study as they have different characteristics than minor version updates. Microsoft Internet Explorer only reports the major version number and omits the minor version number in the user agent string. The often stated reason for this omission is to reduce information leakage and make it harder for an attacker to select a working exploit for the actual browser version in use. As we have

seen drive-by download Web sites trying many different exploits at once, it's unclear how much additional protection this omission really gives. Therefore, based solely on our Web server logs, we cannot determine the update speed of minor versions within the Microsoft Internet Explorer population. However, for four other popular browsers, namely for Google Chrome, Mozilla Firefox, Apple Safari, and Opera, the minor version gets reported.

Some inaccuracies in our measurements are possible, namely due to Web browsers not sending any cookies, which we dropped from our analysis, and those sending a fake user agent string to disguise as a different Web browser. However, the majority of ordinary Internet users are known not to change the default settings of their software. Finally, some geographic regions are covered less as Google has varying popularity in different regions and we only measured visits to Google servers. In first quarter 2009, Google servers performed Web searches for 81% of global users according to Hitlinks [12]. We consider the effects of the mentioned measurement inaccuracies to be negligible, given the global scale and size of our data set.

6 Update Champions and Laggards

We first analyzed the update effectiveness of the four latest non beta releases for Google Chrome, Mozilla Firefox, Apple Safari, and Opera, which were released before mid April 2009, and plotted them against releases of the same Web browser. The plots in Figure 3 show the update effectiveness as percentage of daily active users of a new Web browser version. We plot the shares for the first 21 days after the release and state the Web browser version in the graph legend together with the date of its release. We compared a new release of a Web browser within Web browsers of the same type and the same major version number.

After 21 days of releasing **Google Chrome** 1.0.154.48, an exciting 97% share of active Google Chrome 1.x users were using the latest version of Google Chrome. This is by far the best update effectiveness measured for any of the four investigated Web browsers. It's striking how similar the shares increase for different releases of Google Chrome, indicating the statistical robustness of the process measured. The sudden decrease in the usage share of Google Chrome 1.0.154.46 at the end of the first week after release is explained by the availability of the successor Google Chrome 1.0.154.48. Why Google Chrome is not reaching 100% usage share with new releases even though it silently performs updates without user interaction (and currently without letting the user disable updates) is discussed in our tech report [6].

Mozilla Firefox usage share increases are also strikingly similar across different releases. The usage shares increase slightly faster in the first few days compared to Google Chrome but than flatten out way earlier, never reaching more than 85% usage share for the latest version within 21 days of the release. Frequent checking for updates and the rather obtrusive user prompt to install the new Mozilla Firefox update by restarting the browser most likely help to get users to update earlier. Another factor for such a shift is the actual hour of

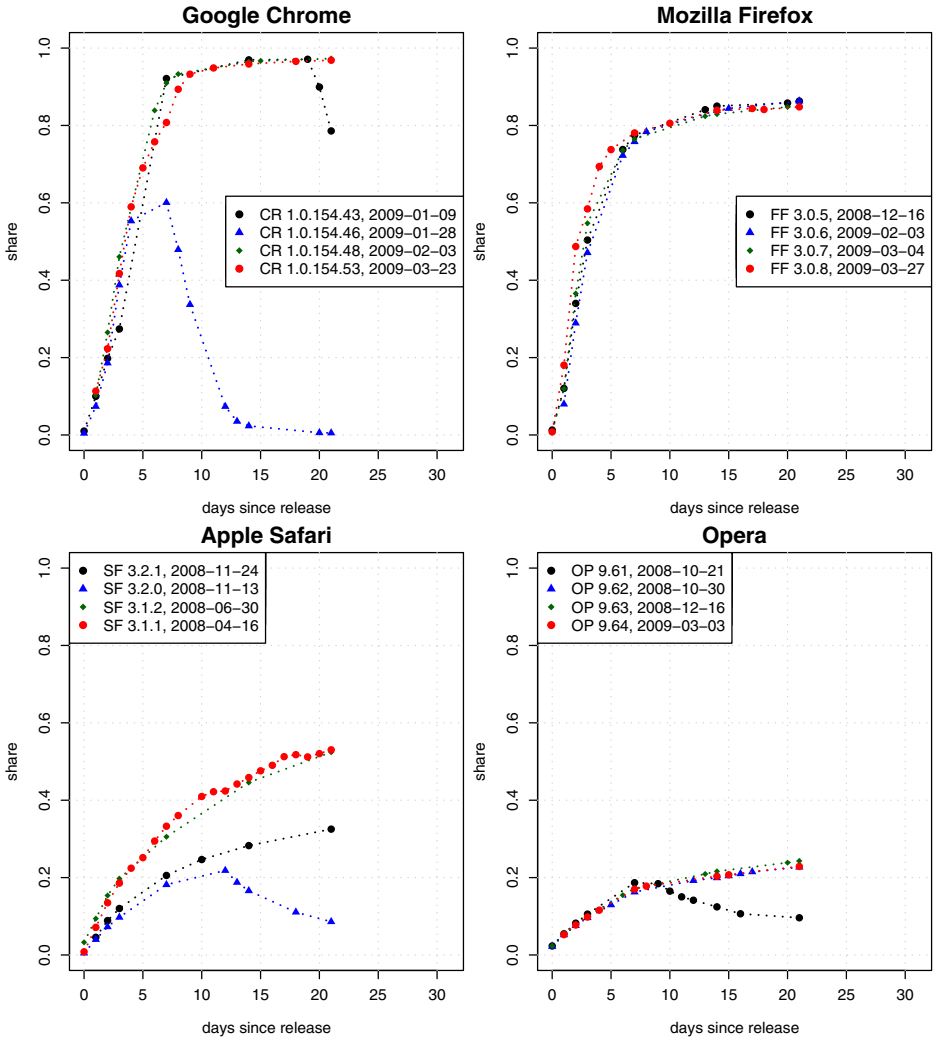


Fig. 3. Usage shares of new browser versions within the same browser major version for the first 21 days after release

the release on the publicized release date, which can also shift results of daily measurements. The fast initial uptake of a new Mozilla Firefox release is a very positive fact of Mozilla Firefox' update mechanism.

A mere maximum 53% share of **Apple Safari** 3.x Web browser users benefit from an update within three weeks of its release. With newer releases of Apple Safari 3.2.x versions, the update effectiveness drops considerably lower. The reason is that Apple put the bar higher to who is eligible for updates to Apple Safari 3.2.x by requiring Mac OS X Tiger 10.4.11 or higher or Mac OS X Leopard 10.5.5 or higher with Security Update 2008-007 installed. Given that

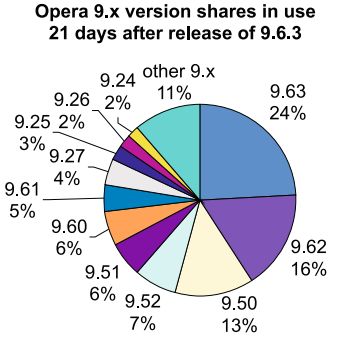


Fig. 4. Distribution of Opera 9.x minor versions (relative to all Opera 9.x versions in use) 21 days after release of Opera 9.63. "Other 9.x" summarizes versions with a less than 2% share each.

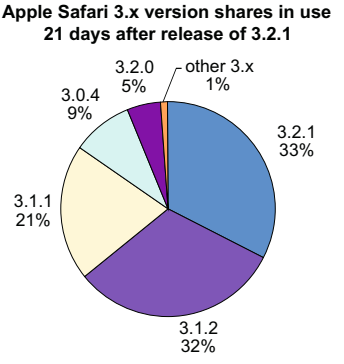


Fig. 5. Distribution of Apple Safari 3.x minor versions (relative to all Apple Safari 3.x minor versions in use) 21 days after release of Apple Safari 3.2.1. "Other 3.x" summarizes versions with a less than 2% share each.

Apple Safari 3.2.1 reaches only 33% on day 21 after release, that's an additional 20% of Apple Safari 3.x users that were left behind since Apple Safari 3.2.x came out. It's not the first time that installation requirements prevent users from updating browsers: users of OS X Panther 10.3, the most recent OS X until OS X Tiger 10.4 was released on April 29, 2005, are limited to Apple Safari 1.3 and Mozilla Firefox 2. Similarly, Windows 9x users have to stick with Mozilla Firefox 2 and Microsoft Internet Explorer 6 and Win 2000 users are limited to Microsoft Internet Explorer 6, the effect of which is measured in [4].

Opera browser users apparently don't update frequently. After three weeks of a new release, a disappointing maximum of 24% active daily users of Opera 9.x have the newest Opera browser installed. It's a pity that 76% of Opera 9.x users currently don't benefit from the security improvements and new features of new Opera versions within three weeks of its release. If some engineering time were spent on increasing update effectiveness instead of working on new features, this would eventually benefit many more users. We also recognize an outlier, namely Opera 9.61, which got replaced after nine days of its release. The upcoming major version of Opera, version 10 now in alpha testing, will update itself automatically as new versions are released [10].

All in all, the poor update effectiveness of Apple Safari and Opera gives attackers plenty of time to use known exploits to attack users of outdated browsers. Figure 4 shows the version mix of Opera 9.x browsers in use relative to all Opera 9.x browsers, 21 days after Opera 9.63 was released. This version reached the highest measured update effectiveness of the last four Opera releases with 24% usage share. Similarly, we give in Figure 5 the shares of Apple Safari 3.x minor versions relative to all Apple Safari 3.x versions in use, 21 days after Apple Safari 3.2.1 got released.

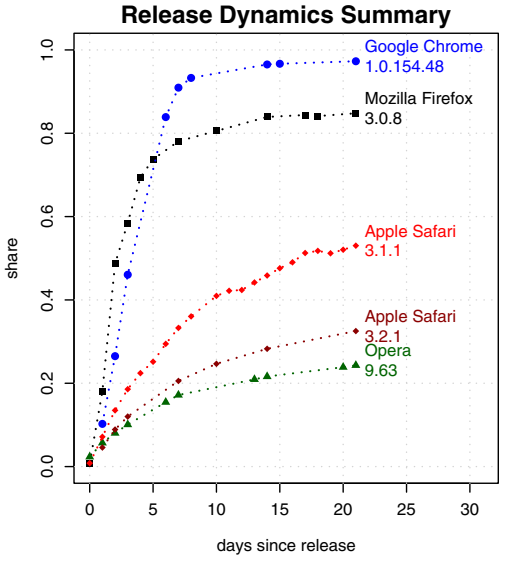


Fig. 6. Shares of active users visiting Google Web servers with the indicated Web browser versions measured relative to users of the same browser major version over 21 days after each update was released

The best performing version of each browser vendor was then selected and plotted in Fig. 6. For Apple Safari, we took the best 3.1.x and 3.2.x versions to illustrate how additional system dependencies can degrade effectiveness.

7 Update Effectiveness by OS

Given that the logged HTTP user agent string also contains the operating system used by the Web browser, we decided to split the active user base by operating system in addition to browser type. The result of this analysis is plotted in Figure 7 and it reveals interesting differences in the update effectiveness by operating system.

For Firefox 3.0.8, update effectiveness is almost identical for Windows Vista (87% of active users are updated on day 21 after release), Windows XP (85%) and Mac OS X (83%). The reason is that the update mechanism is independent of the operating system as it is built into the browser on these platforms. However, on Linux and especially on Unix, a significantly slower update speed is visible. We attribute this to two facts: On one side, manual installation of new Firefox versions using tar archives are a hurdle and require the user in the loop. On the other side, many users rely on the distribution of Web browser updates through the Linux package management system instead of using the browser's built-in update mechanism (e.g. the Debian package distribution provides rebranded Mozilla Firefox browsers under the name Iceweasel due to Mozilla's

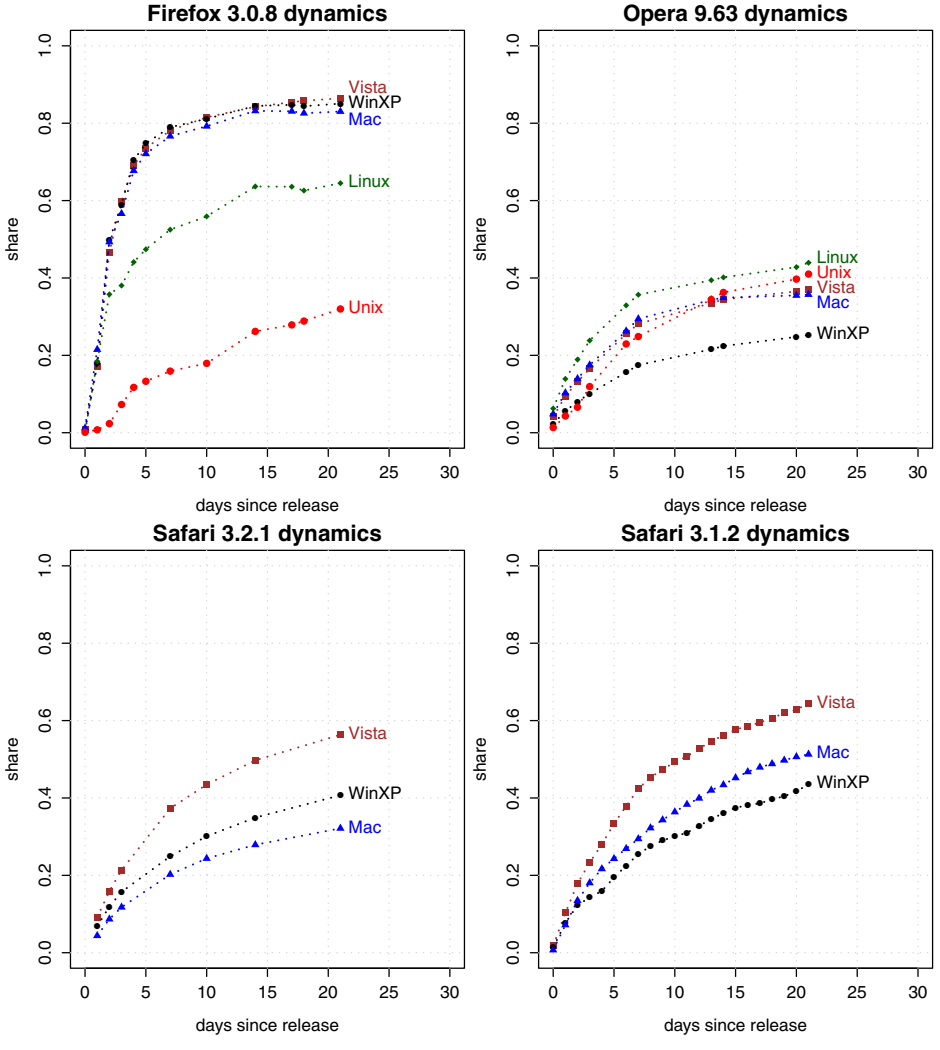


Fig. 7. Usage shares of new browser versions by operating system within the same browser major version for the first 21 days after release

licensing restrictions). There's typically a time lag for an update to be included as a new package because the package management system maintainer needs to verify the package with thorough checks for compatibility and security, which take some time. Furthermore, some organizations run an inhouse package mirror, which can introduce additional delays before an update is available and distributed to local machines. The frequency of update checks by the local package management system is set by the system administrator and finally, the user needs to restart the browser once an update is installed. Our measurements

indicate that the browser's built-in update mechanism is significantly more efficient to protect a larger user base while requiring less time for update distribution to target systems compared to the Linux and especially Unix package management systems.

Opera 9.63's update performance is best on Linux (44% on day 21 after release), closely followed by Unix, Windows Vista and Mac OS (36%). The user base on Windows XP is updating the slowest (26%). We attribute this to the cumbersome manual update installation process that is basically identical with a fresh installation of Opera. Our measurements clearly illustrate the negative performance effect resulting from relying on the user for update installation and the lack of update automatism in Opera 9.x. For Linux, Opera also provides updates through Linux' package management system. However, in order to get e.g. the Debian packages, the administrator needs to configure the extra Debian package source `http://deb.opera.com/opera` as it is not included in the default package sources. Opera 10.x, which was released on 1st of September 2009, finally has a built-in auto update mechanism.

For Safari 3.2.1, we found that the Safari user base on Windows Vista (57% on day 21 after release) updates the fastest, followed by Windows XP (41%). Surprisingly, our measurements indicate that Safari users on Windows are safer than Safari users on Mac OS X. Given that Safari 3.2.1 users on Mac OS X update the slowest (32%), the time window of known vulnerabilities being present in the actively used Safari browser version is longer for Mac OS X users of Safari compared to Windows XP users of Safari. A plausible explanation is that Safari 3.2.1 on Windows did not require a security update for the OS for its installation to succeed whereas this was the case for Mac OS X users. For the earlier version Safari 3.1.2 users on Windows Vista (65%) got updated most efficiently, followed by Mac OS X (52%) and Windows XP (44%) users. However, Safari 3.1.2 for Windows was released 10 days earlier than the Mac OS X version and consequently, Windows users again got the Safari security update first.

8 Discussion of Patching Strategies

Patch management in general (not only of Web browsers) is a complex undertaking – especially in the context of large, business critical infrastructures typically found in organizations. Thus, in the last decade the increasing number of patches has led corporate customers request software vendors for a more predictable schedule to plan the patching of their complex IT environments. As a result, several big software vendors like Oracle or Microsoft today typically release all their security patches on a predictable schedule (e.g. every month or quarter). There are no out-of-band patches except for rare, highly critical emergency ones. We consider it suboptimal if patches for heavily attack exposed and highly prevalent applications such as Web browsers and plug-ins with more than a billion users worldwide, are delayed due to the need for a fixed patch schedule for some business users. A fixed patch schedule mainly benefits the patch management processes of larger corporations - organizations which are

typically better protected against Internet threats than the masses of individual users. Based on our measurements and the evolution of the threats towards end-users we suggest that software vendors release patches for attack exposed applications, such as Web browsers and plug-ins, as soon as they are available - while keeping a patch schedule for less attack exposed applications. By default the Web browser should be set to install updates silently, with the option for power users to change this configuration. This would optimally protect the large masses of users while giving professionals the options needed to customize their environment. We believe that there is room for a better trade-off to benefit overall security.

Could a silent update mechanism also pose a security problem itself? Due to the nature of the update being silently deployed with most users unaware of the software change, there are less other parties besides the vendor that test such an update before deployment compared to semi-automatic or manual updates. An update could break a software and prevent the user from using it (i.e. cause denial of service). Therefore, it's important that the silent update mechanism is separate from the software it updates to allow for installing another non faulty update as soon as available. A user initiated rollback to a previous software version (and then blocking the silent update for a given time) would give the user some control back and allow for continued operation even after a faulty update was distributed.

9 Conclusions

Our global measurements have empirically proven that Web browser Google Chrome's silent auto update mechanism is the most effective compared to those of Mozilla Firefox, Apple Safari, and Opera. With 97% latest version share among daily active users three weeks after a new release, it clearly reached the best result. We think that Google Chrome's update effectiveness could be further improved by gently notifying the user about the needed browser restart for changes to take effect after the installation of a new release is done. Furthermore, our results show that Mozilla Firefox' update mechanism has the fastest initial update effectiveness in the first five days after release. Being able to deploy security patches for Web browsers quickly to all end users greatly increases end system security as known vulnerabilities can't be exploited anymore.

In the case of Apple Safari 3.2.1, we have noticed that coupling browser and operating systems and consequently requiring the user to have a recent operating system patch level in order to be eligible to install a browser update should be avoided. Apple left an additional 20% of Apple Safari 3.x users behind with an outdated browser version compared to the previous update to Apple Safari 3.1.2, which did not have these requirements.

Given that today's best performing update mechanism, Google Chrome's Updater Omaha [9], was recently open sourced and is free to use for anyone, we encourage others to try it out for their own software. With silent updates, the user does not have to care about updates and system maintenance and the system stays most secure at any time. We think this is a reasonable default for

most Internet users. Furthermore, silent updates are already well accepted for Internet Web applications, so it's only logical to extend this to the client platform that runs the Web apps. For server operating systems and high availability applications like databases, silent updates prevent the system administrator from carefully testing a system change before deployment to production. In such environments, enforcing silent updates is currently not a viable option. While applying security updates silently also to client operating systems and desktop applications other than Web browers would be beneficial to minimize the attack surface further, power users currently prefer to keep control over which exact software is running on their systems. With new Web centric operating systems like Google Chrome OS with minimal need for system administration by the user, this might radically change.

Acknowledgements. We would like to thank Mark Larson, Carl Nygaard, Mike Smith, and Linus Upson (listed in alphabetical order) and the anonymous peer reviewers of CRITIS 2009 for their feedback on this paper.

References

1. Frei, S., Duebendorfer, T., Ollmann, G., May, M.: Understanding the Web browser threat. Technical Report 288, TIK, ETH Zurich. Presented at DefCon 16, August 2008, Las Vegas, USA (June 2008), http://www.techzoom.net/insecurity-iceberg
2. Frei, S., Schatzmann, D., Plattner, B., Trammel, B.: Modelling the Security Ecosystem - The Dynamics of (In)Security. In: Workshop on the Economics of Information Security (WEIS), UK (June 2009) http://weis09.infosecon.net/, http://www.techzoom.net/security-ecosystem
3. Finjan. How a cybergang operates a network of 1.9 million infected computers. MCRC Blog - 2009 (April 2009), http://www.finjan.com/MCRCblog.aspx?EntryId=2237
4. Frei, S., Duebendorfer, T., Plattner, B.: Firefox (In)security update dynamics exposed. SIGCOMM Comput. Commun. Rev. 39(1), 16–22 (2009), http://doi.acm.org/10.1145/1496091.1496094
5. Google Chrome Web browser, http://www.google.com/chrome
6. Duebendorfer, T., Frei, S.: Why Silent Updates Boost Security. Technical Report 302, TIK, ETH Zurich (May 2009), http://www.techzoom.net/silent-updates
7. NIST. National Vulnerability Database (NVD), http://nvd.nist.gov
8. Common Vulnerability Scoring System (CVSS) Calculator, http://nvd.nist.gov/cvss.cfm?calculator&version=2
9. Omaha, the open source Google Updater, http://code.google.com/p/omaha/
10. Opera, http://my.opera.com/desktopteam/blog/index.dml/tag/auto-update
11. Microsoft Security Bulletin MS08-078 (December 2008), http://www.microsoft.com/technet/security/bulletin/ms08-078.mspx
12. NetApplications.com. Search Engine Worldwide Market Share (March 2009), http://marketshare.hitslink.com/report.aspx?qprid=4

State-Based Network Intrusion Detection Systems for SCADA Protocols: A Proof of Concept

Andrea Carcano[2], Igor Nai Fovino[1], Marcelo Masera[1], and Alberto Trombetta[2]

[1] Joint Research Centre, Institute for the Protection and Security of the Citizen, via E.Fermi 1, 21027, Ispra, Italy
[2] University of Insubria, via A.Dunant, 21100 Varese, Italy

Abstract. We present a novel Intrusion Detection System able to detect complex attacks to SCADA systems. By *complex* attack, we mean a set of commands (carried in Modbus packets) that, while licit when considered in isolation on a single-packet basis, interfere with the correct behavior of the system. The proposed IDS detects such attacks thanks to an internal representation of the controlled SCADA system and a corresponding rule language, powerful enough to express the system's critical states. Furthermore, we detail the implementation and provide experimental comparative results.

Keywords: Security, SCADA systems, critical infrastructures, IDS.

1 Introduction

Modern *critical infrastructures* (e.g. power plants, water plants etc.) largely use ICT technologies in order to provide new services and offer new features. Several of the maintenance and management operations related to such installations are conducted remotely taking advantage of public networks (i.e. Internet). If, on one hand, this has contributed to improve many operations in such systems, allowing at the same time to reduce the maintenance costs (less operators, unified control centers etc.) and to boost efficiency, on the other hand it has exposed such critical installations to new sources of possible threats. In fact the interconnection of critical systems through Internet has opened new security holes. The ICT protection of critical infrastructures assumes in this context an extremely relevant role. Several studies [1] [2] [3] proved that modern industrial critical infrastructures are, on average, exposed to the traditional computer attacks and threats. Common ICT security technologies (e.g. ICT Firewalls, Intrusion Detection Systems, antivirus etc.) are usually apt for the avoidance of the majority of these threats. Unfortunately, industrial critical installations are composed not only of traditional PCs running conventional, corporate software, but are composed of dedicated systems, programmable logic controllers (PLCs, Remote Terminal Units (RTU) as well, running ad-hoc software and communicating through dedicated communication protocols (Modbus, DNP3, ProfiBus etc.) under real-time

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 138–150, 2010.

constraints. As showed by Carcano et al. [2], at the present time, traditional ICT security technologies are not able to protect in an adequate way production systems from ad-hoc Supervisory Control and Data Acquisition SCADA attacks. We defend that a new set of dedicated ICT security technologies need to be designed in order to protect industrial critical installations. In this paper, we focus our attention on intrusion detection technologies. In the field of *SCADA systems*, as will be described in the following section, IDS techniques are extremely embryonic and far to be considered as mature. The most advanced solutions are still at the level of *single packet* analysis, in which an alert is raised if a malicious packet is sent to a PLC/RTU. According to such strategies it is possible to detect simple attacks in which the main scope is to interfere with the state of a single SCADA slave. However, more complex, coordinated attacks, in which allowed commands are sent to different SCADA actors in order to influence the state of the whole system, cannot be detected with this kind of approach. We present a first prototype of a *State Based Network Intrusion Detection System* tailor-made for analyzing Modbus traffic, aimed at identifying complex attacks which might interfere with the state of an entire SCADA installation.

Motivating Examples. Consider a system with a pipe in which flows high pressure steam. The pressure is regulated by two valves (1 and 2). An attacker able to send packets on the process network, sends a Modbus packet to the (PLC controlling) valve 1 in order to force its complete closure, and a command to the (PLC controlling) valve 2 in order to maximize the incoming steam. It is evident how such commands, when considered locally, will result perfectly legal to a traditional IDS, while, altogether will bring the system to a critical state.

In order to mitigate such risk, it is necessary to provide the IDS a detailed, explicit knowledge of the SCADA system under analysis (components, commands and critical states). We remark that the IDS checks *all* the network traffic and not only the packets generated by the SCADA master. Thus, the IDS is able to detect (possibly) forged packets as long as the carried instructions yield the system in a critical state.

2 Related Works

Nowadays, Intrusion Detection is a well established field of research. The basic idea, presented in the mid-eighties (see for example [4]), is to search evidences which indicate that a malicious attack is in act in a certain time window. Intrusion detection techniques can be clusterized according to different parameters:

- **source of information:** the evidences can be derived from the observation of the local behavior of a set of hosts (Host based IDS), or from the analysis of the network traffic generated by a set of hosts (*Network* based IDS, or NIDS)
- **Detection technique:** the detection of the evidences can be performed through some pattern matching between a set of signatures describing known malicious operations and the reality observed (signature based detection), or identifying the deviations from the standard behavior of the system under analysis (anomaly detection)

In this work, we concentrate our attention to NIDS techniques. The reason is the following: the main components of a SCADA system are PLCs and RTUs, which have usually low computational and memory resources. Host based IDS sensors, conversely, need to be installed on the systems (PLCs, RTUs in our case) to be analyzed and require a certain amount of memory and computational power on the host machine. This makes such techniques not suitable to monitor SCADA devices. On the other hand, NIDS sensors, in order to perform their work need only to be installed on a machine having the full access to the network segment to be monitored. In other words, they are seamlessly integrated with respect to the SCADA system process profile. In a field network in which interferences and performance degradation might have catastrophic effects (consider for example the SCADA system of a chemical or nuclear plant), such features of the NIDS make them the most suitable approach in this context.

Typical NIDS architectures are composed of a number of distributed sensors which analyze the traffic flowing through a network, searching for attack signatures and anomalies. In the case of SCADA systems, such NIDS should be able to understand and analyze industrial communication protocols like Modbus, DNP3, Profibus etc. In their current implementations, these protocols (originally conceived for serial communication) have been ported over TCP/IP and embedded into the payload of a TCP packet. Traditional NIDS, like Snort [5], were unable to understand such "application level" protocols. Only recently a set of ad-hoc rules [6] have been released in order to detect some attacks over these protocols. Such signatures can be clustered into the following categories:

- Unauthorized Modbus Users rules
- Modbus Protocol Errors rules
- Scanning detection rules

Roughly speaking, a traditional NIDS provided with such rules could be able to identify very primitive, single packet-based attacks, in which the attacker just sends a forbidden packet to a target PLC, or uses rare commands. However, as showed in [1], nowadays SCADA attacks can be extremely complex, and are rarely composed of just one step (i.e. the exploit of a single vulnerability). Conversely, the trend is towards attacks consisting of a complex sequence of simpler (we say *atomic*) steps. In this case the required capability is the identification of complex and dangerous attacks, by the analysis and correlation of different, low-risk, atomic operations. This is known in literature as *attack correlation*.

In the scientific literature on IDS systems for traditional ICT, exist some examples of similar approaches. In particular, Gross et al. [7] proposed a mechanism for collaborative intrusion detection ("selecticast") which uses a centralized server in order to dispatch among the ID sensors information about activities coming from suspicious IP addresses. At the same way Yegneswaran et al. [8]introduce the concept of *Distributed Overlay for Monitoring InterNet Outbreaks*. These approaches are, of course, useful in order to provide a broader picture of the suspicious events happening in the whole monitored system to each intrusion detection sensor. However it does not provide any kind of specific technique allowing to identify high level and complex malicious actions especially

in relation with industrial process systems. Ning et al. [14] propose a model which aims at identifying causal relationships between alerts on the basis of prerequisites and consequences. Similarly, the approach proposed by Cuppens and Miege in [9] adopts pre and postconditions; moreover, it includes a number of analysis phases as: alert clustering, alert merging and intention recognition. In the intention of the authors, such approach should facilitate the automatic generation of correlation rules. Unfortunately, this technique, in several situation, can generate a number of spurious correlation rules that might increase the noise in the ids alerting system. In each of the presented techniques the correlation is based on the capability of identifying "malicious actions". In other words, the alerts related to such malicious actions became the feeds of the correlation systems. However, in industrial system, as explained in the example provided in the section 1, a chain of apparently licit events could bring the system into an unwanted, critical state. For that reason, we believe that an IDS for process networks, should be able to correlate also licit events in search of malicious final states.

3 A State-Based SCADA IDS Architecture

On the light of the considerations previously made, in this section we present a new approach to intrusion detection for SCADA systems, based on the concept of system knowledge base and system state analysis, specifically tailored for the context of complex critical infrastructures (i.e. Power Plant and Power grid systems, Water grids etc.). As we have already claimed, the classical "Snort style" signature based approach is too primitive to be used in a context critical like an industrial system. Such an approach provides protection only against single packet-based attacks leaving opened all the issues related to more complex and subtle attacks (like the example described in the previous section). On the other side, pure statistical anomaly detection might amplify the well known "false positive" problem of such set of techniques, when considering the number of different field devices usually used in a process/field network. The problem is then how to detect known and unknown attacks while limiting at minimum the risk of false alerts. This problem still is – of course – one of the more active matter of research in the IDS field. The area of *industrial processes*, although extremely complex from an architectural point of view, has the advantage to be extremely structured and well defined. Every process is well known, described and documented; every state of the system has been analyzed according to the common safety and reliability methodologies. In other words, in these systems, the critical states (i.e. the set of system configuration which might cause system stops, damages etc.) are well known.

Starting from the assumption that every attack which aims at damaging a SCADA system at the level of the field network has, as side effect, the transition of the system state from a *secure state* to a *critical state*, the description of the *System Critical States* (SCS) becomes a very relevant source of information which can be effectively used in an intrusion detection process.

In our approach, by the use of an already developed and consolidated system description methodology [10,11,12,13], the *system knowledge* is decomposed in term of components (PLCs, RTUs, SCADA Masters), information flows, critical states and vulnerabilities associated to the components. All this knowledge base is used to keep track of the current state of the SCADA system under analysis. In other words, the IDS, while analyzing the packets in search for known signatures (snort style signatures), keeps updated a digital representation of the system physical state. We will refer thereon to such system's representation as the system *virtual image* . In this way, every time an allowed – IDS recorded – command brings the virtual image of the system into a well-known, described *critical state* an alert is raised. In this way, complex SCADA attacks (like the example described in the previous section) will be identified as well. Figure 1 shows the high level description of the our IDS for SCADA systems.

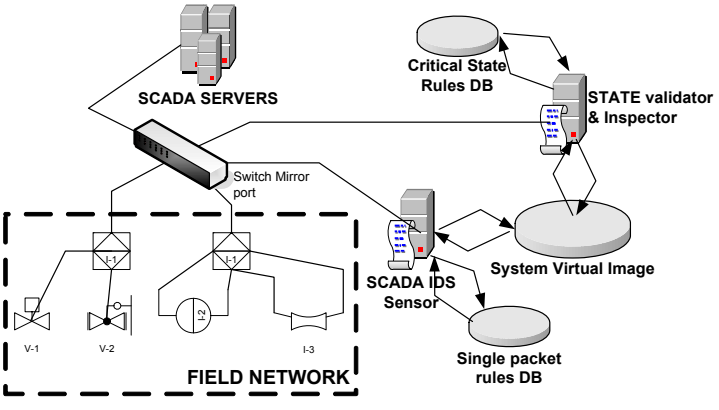


Fig. 1. High level SCADA IDS architecture

From a logical point of view, we identify five logical element in the IDS architecture:

- **SCADA Protocol Sensor** (SPS): it is basically the equivalent of a Snort sensor, with the advantage of being able to directly support commonly used SCADA protocols, like Modbus and DNP3. It receives the mirror of all the traffic to and from the field network directly from the SPAN port of the field network switch, it builds up the SCADA protocol flow and analyzes it in search of matching within the Single Packet Rules DB. Moreover, it uses the information retrieved in order to keep updated the system state.
- **Single packet rules DB** (SPDB): It contains all the signatures of the known single packet-based attacks on SCADA protocols.
- **System Virtual Image** (SVI): It contains the representation of the state of the system under analysis. That is to say, the SVI is a collection of data structures representing the different PLCs and RTUs of the monitored system, with their memory registers, coils, inputs and outputs.

- **State Validator and Inspector** (SVAL): It checks the consistency of the System Virtual Image and eventually queries the field devices to keep synchronized the SVI with the physical system state. Moreover, it verifies, according to the rules stored into the Critical State Rules DB, if the system is entering into a critical state, and in that case, raises an alert, notifying the pool of packets which have contributed to bring the system in such critical state as well.
- **Critical State Rules DB** (CSRDB): it contains the rules describing all the critical states of the system under control.

From an operative point of view, the adopted strategy is the following:

1. SPS analyzes all the traffic flowing through the field network, re-constructs the SCADA protocol flows and according to the rules specified into SPDB, identifies the low level malicious packets raising an alert if it's the case. Moreover, it sends all the reconstructed, operative commands to SVAL, in order to update the virtual image of the system.
2. SVAL in the mean time, performs basically two operations:
3. It keeps updated the SVI taking advantage of the packets received from the SPS, and performing periodical queries to the field devices in order to check their state and guarantee the SVI consistency.
4. Every time a new packet is received from the SPS, it verifies, according to the rules stored into the CSRDB, the state of the system, raising an alert if the process is entered into a critical state

Three points are relevant to our approach: the SPS implementation, the rule language for describing the low-level signatures and the critical states, and the implementation of the system virtual image. In the following we describe in detail such points and illustrate the experimental results of our implementation.

4 Rules Languages

In our approach, we have adopted two detection techniques: (i) a single packet signature based technique (ii) a State Analysis technique. In both the two cases we need to define a language allowing to express the analysis rules (to describe malicious packets in the case (i) and to describe critical states in the case (ii)). In this section we provide a brief definition of the two languages.

4.1 Packet Language

As it is common in every signature language definition, our rules have the form

$$signature \rightarrow action$$

where the *signature* express exactly the content of a target malicious Modbus packet, while *action* represents the kind of operation the IDS has to execute if a signature matches with an analyzed packet. In our case we define for this

prototype two possible actions: *Log*, which will force the IDS to store into a log file the matching packet, and *Alert*, which will force the IDS to raise an Alert if a packet matches the related signature.
More in details, in our language a *packet signature* has the following format:

$$\{Source|Destination|Port|Function|Payload\}$$

Where:

- **Source** is the IP address of the packet sender.
- **Destination** is the IP address of the packet receiver (e.g. the IP address of a PLC).
- **Port** is the TCP (Server) destination port used for the communication.
- **Function** is the Modbus function invoked by the sender.
- **Payload** contains all the data/parameters required by the invoked Modbus function.

The packet signature contains only one port field because according to the Modbus specifications only the destination port is relevant for the IDS analysis. In fact, in a standard Modbus TCP/IP communication the server port 502 is the standard listening port. It is mandatory to listen by default on that port. When the Master needs to exchange data with a remote server, it must open a new client connection with a remote Port 502 and a local random port that must be higher than 1024 and different for each client connection.

We recall here that our IDS has been conceived in order to be able to reconstruct the Modbus protocol data-flow. This fact implies that, differently from other general purpose IDS like Snort, in which in order to analyze Modbus packets it is needed to write extremely complex rules which analyze the content of a generic TCP payload, in our case, the rule language can directly refer to Modbus protocol field and functions, making a lot easier the process of rule writing.

4.2 Critical State Language

We now describe the rule language we employ to detect licit packets which might put the system into a critical state. We recall that such signatures depend on state of the overall system architecture, represented as described in Section 3. A rule in our language has the form $condition \rightarrow action$, where *action* may represent an alert, a log of (possibly part of) the payload of the inspected packet that triggered the rule or a look-up to another rule. The other part of the rule, *condition*, is a boolean formula, composed of conjunctions and/or disjunctions of predicates describing what values can be assumed by the different PLCs' components.

The PLC's elements taken into account by our rule language are coils C, registers R, digital inputs DI, digital outputs DO, analog inputs AI and analog outputs AO.

The fundamental difference of our rule language with respect to other traditional IDS rule languages lies in the fact that the predicates (upon which the

condition that triggers the corresponding rule is formed) are defined over *different* PLCs (hence, a given rule is triggered depending on the payloads of *different* packets). This is possible given the architecture of the IDS proposed in this work. In fact, upon inception of a packet, the IDS updates the SCADA system state, changing the parameters of the PLC (or PLCs) to which the packet is addressed, according to the information contained in the packet payload.

Then, the IDS checks whether some PLCs' configuration as represented in the resulting state triggers a rule. If this is the case, the corresponding action – prescribed by the rule – is performed. We now present the rule language using standard BNF notation:

$\langle rule \rangle := \langle condition \rangle \rightarrow \langle action \rangle$
$\langle condition \rangle := \langle predicate \rangle \mid \langle predicate \rangle \langle conn \rangle \langle predicate \rangle$
$\langle predicate \rangle := \langle term \rangle \langle relation \rangle \langle term \rangle$
$\langle term \rangle := \langle PLCName \rangle \mid \langle value \rangle$
$\langle PLCName \rangle := PLC \langle number \rangle . \langle comp \rangle \langle number \rangle$
$\langle action \rangle := Alert \mid Log \mid Look \langle rule \rangle$
$\langle conn \rangle := and \mid or$
$\langle relation \rangle := \leq \mid \geq \mid < \mid > \mid =$
$\langle comp \rangle := C \mid R \mid DI \mid DO \mid AI \mid AO$
$\langle value \rangle := 0 | ... | 2^{16} - 1$

Having in mind the example shown in Section 1, consider the following rule stating that if coil $C23$ of $PLC1$ has value 0 and coil $C17$ of $PLC2$ has value 1 (corresponding, respectively, to open Valve 1 and close Valve 2), then the IDS performs an *Alert* action of the last packet that changed the state

$$PLC1.C23 = 0 \, and \, PLC2.C17 = 1 \rightarrow Alert$$

Thus, a packet addressed to $PLC1$ containing a command for switching coil $C23$ to 0 – given that the other condition stated in the rule is satisfied – will trigger an *Alert* action. Again, note that the switching request (local to $PLC1$) could be a perfectly legal one, but it could become critical when other, non-local conditions are tested.

5 Implementation Details

Our SCADA IDS prototype is implemented according to a modular approach, in which each functional component constitutes a separate module. In this way, in a following stage of our research work, it will be easier to add new modules in order to improve the features and functionalities of our Intrusion Detection System.

The prototype is based on three modules, implemented in C♯. The first module, named *Load System* (LS) is in charge for the initialization of the system virtual image through a configuration XML file. The basic information required at the beginning is:

- The PLC's brand and model
- The numbers of registers and coils for each PLC
- The process network topology (in our prototype we take into account the Multidrop process network topology.)
- The default PLC register value or coils different from 0
- The PLC number, registers and coils that have to be monitored directly

The LS module uses this information for the creation of the SCADA system virtual image, mirroring the PLCs topology and configurations. Registers and coils values for each slave are stored in a corresponding data structure and used by the modules in the virtual image analysis process. In particular each different data type in the PLC can be referenced by the array index.

The most important method adopted from the Loader Module are:

- **Update** sent a reading request to all the slave connected to the master and update the FW system model.
- **UpdatebyFunction** starting from the payload it changes the system model thorough the Protocol Builder.

The second module is called *State Controller* (SC). It keeps updated the system virtual image by sniffing the traffic between the process network and the field network and changing as consequence the system virtual image. After reading a Modbus packet from the master to a slave, the SC module takes note of all the writing functions (from the master to the slave) and of the reading response (from the slave to the master) that are relevant for a system virtual image update. Afterward, the SC module performs such update.

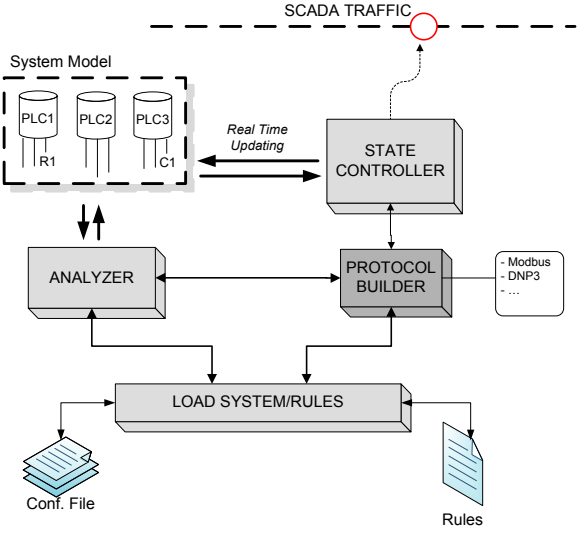


Fig. 2. IDS implementation

In order to keep a tight synchronization between the system virtual image and the physical system, the SC module includes a master simulator able to send the reading command to the PLCs and update the registers and coils which are not monitored by the SCADA system specified in the configuration XML file.

The third Module, called *Rules Analyzer* (RA), monitors the Virtual System Image in search of transitions to critical states. This is done by scanning the updated virtual image and checking whether at least one of the critical state rules (as described in Sectionsec:rulelang) is triggered. This part is implemented using synchronized methods in order to control the data access. The Protocol Builder (PB) contains all the Methods and the parameters used by the others models for interpret the IDS rules and generate and update the system.

The prototype has been originally implemented in C♯ (MS.NET framework version 2.1) in a MS-Windows environment and then imported in a standard Linux environment (Ubuntu 9.04).

6 Comparative Tests

In order to verify the efficiency and validity of our SCADA IDS, we have performed several comparative intrusion detection tests in a protected environment located in our laboratory, developed in collaboration with a power company, which entirely reproduces the network, hardware and software environment of a typical power plant. We have compared the detection results obtained by using our SCADA IDS with the results obtained with Snort[5].

More in detail, we have performed two kind of test:

- Single Packet Signature based Detection
- Critical State Detection

From an architectural point of view, we adopted for the field network a classical Multidrop topology composed by a Master and six PLCs with the same characteristics. The internal PLC's data structure is composed by:

- 500 Discrete Input(1=ON, 0=OFF).
- 500 Discrete Output or Coils (1=ON, 0=OFF).
- 120 Input registers
- 250 Holding registers

All this information is stored in the XML configuration file loaded by the IDS before starting to analyze the SCADA traffic.

6.1 First Phase

In the first phase we have tested the single packet detection strategy described in Section 5 by inserting a rule allowing to block all the packets apparently sent by the master to a target slave (IP address 10.0.2.2) that tries to switch the values of ten contiguous coils to the state ON(1) starting from coil address 20, and a set

of rules allowing to write in a log file all the *"writing holding register function"* sent apparently by the master to another slave (IP address *10.0.2.5* (PLC5))

$$10.0.0.1|10.0.2.2|502|15|20, 10, 2, 255, 3 \rightarrow deny$$

$$10.0.0.1|10.0.2.5|502|06|..... \rightarrow log$$

$$10.0.0.1|10.0.2.5|502|16|..... \rightarrow log$$

In order to obtain a situation similar to a real SCADA system we sent a huge amount of different Modbus commands (rate 1/500ms), randomly injecting in the flow 10 commands which tries to write the PLC5's holding register and 2 commands trying to write ten contiguous coils of the PLC2 starting from the coil number 20. Both the SCADA IDS and *SNORT* system were able to identify the malicious Modbus requests without any delay.

6.2 Second Phase

In the second phase we have defined three typical critical states, in order to show how our Intrusion Detection System can raise an alert also when a set of typical Modbus (apparently safe) request is forged with the scope of damaging the critical infrastructure.

In the following some rules used are showed.

$$PLC1.C2 = 1 and PLC1.C12 = 1 and PLC4.C7 = 0 and PLC4.C8 = 0 \rightarrow Alert$$

With this rule we want to identify the attempt of closing two valves (PLC4.c7 and PLC4.c8) while other two are opened (PLC1.C2, PLC1.C12).

$$PLC1.R1 > 2000 and PLC3.C1 = 1 and PLC4.C20 = 1 \rightarrow Alert$$

In this situation we describe how register value can't assume values below 2000 when the value of two coils is 0.

$$PLC1.R1 > 45 and PLC3.C1 = 1 \rightarrow Alert$$

The last rule allows to monitor the value usually not checked by the Master but nevertheless important for the critical infrastructure. In particular, the IDS will rise an alert when the coils C1's value is set to ON and the value of the register R1 reach the value 45.

Note again that when considered in isolation, packets satisfying all the above conditions do not raise any action from a SNORT style IDS. On the other side, thanks to the deployment of the virtual image system and the corresponding rules denoting the critical states, our IDS is able to discover all the potential dangerous situations.

7 Conclusions

The ICT security of industrial systems is gaining great importance in the context of their criticality for society at large. Due to the traditional peculiarities of the process systems, general purpose ICT security mechanisms cannot be considered fully effective. There is then an urgent need for a new generation of security techniques tailored for such systems. In this paper we have presented a first prototype of IDS specifically designed for SCADA architectures. The prototype, able to analyze Modbus packets, adopts two detection strategies: (a) A single packet signature based strategy, allowing to detect illicit packets sent to PLCs and RTUs, and an innovative State Based Intrusion Detection technique, allowing to keep track of the intrustial system state, and to identify if a set of licit Modbus commands sent to the field devices is able to bring the system into a critical state. The comparative tests performed showed how the presented IDS is able to detect at least the same single packet based attacks of Snort, while completely overpass the Snort capabilities in detecting complex attack scenarios based on licit Modbus packet chains. Finally, we presented a rule language specifically designed in order to describe Modbus signatures and field device states. For the future we are planning to extend the analysis capabilities to the other field communication protocols (DNP3, Profibus, Fieldbus etc.), improve the state analysis technique and the state description language.

References

1. Nai Fovino, I., Masera, M., Leszczyna, R.: ICT Security Assessment of a Power Plant, a Case Study. In: Proceeding of the Second Int. Conference on Critical Infrastructure Protection, Arlington, USA (March 2008)
2. Carcano, A., Nai Fovino, I., Masera, M., Trombetta, A.: Scada Malware, a proof of Concept. In: Proceeding of the 3rd International Workshop on Critical Information Infrastructures Security, Rome, October 2008, pp. 13–15 (2008)
3. East, S., Butts, J., Papa, M., Shenoi, S.: A Taxonomy of Attacks on the DNP3 Protocol. In: Proceeding of the Third Int. Conference on Critical Infrastructure Protection, Hannover, NH, USA (March 2009)
4. Denning, D.E.: An Intrusion-Detection Model. IEEE Transactions on Software Engineering SE-13(2), 222–232 (1987)
5. Roesch, M.: Snort -Lightweight Intrusion Detection for Networks. In: Proceedings of LISA 1999: 13th Systems Administration Conference, Seattle, Washington, USA, November 1999, pp. 7–12 (1999)
6. http://www.digitalbond.com/index.php/research/ids-signatures/modbus-tcp-ids-signatures/ (last access 9/04/2009)
7. Gross, P., Parekh, J., Kaiser, G.: Secure Selecticast for collaborative Intrusion Detection systems. In: Proceedings of the International Workshops on DEBS (2004)
8. Yegneswaran, V., Barford, P., Jha, S.: Global Intrusion Detection in the Domino Overlay System. In: Proceedings of the 11th ANDSSS Conference (2004)
9. Cuppens, F., Miege, A.: Alert correlation in a cooperative intrusion detection framework. In: Proc. Security and Privacy (2002)

10. Nai Fovino, I., Masera, M.: A service oriented approach to the assessment of Infrastructure Security. In: Proceeding of the First Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection, Dartmouth College, Hanover, New Hampshire, USA, March 2007, pp. 19–21 (2007)
11. Nai Fovino, I., Masera, M.: Emergent Disservices in Interdependent Systems and System-of-Systems. In: Proceeding of the IEEE Conference on Systems, Man and Cybernetics, Taipei, October 2006, pp. 8–11 (2006)
12. Masera, M., Nai Fovino, I.: Models for security assessment and management. In: Proceeding of the International Workshop on Complex Network and Infrastructure Protection (2006)
13. Nai Fovino, I., Masera, M.: Modelling Information Assets for Security Risk Assessment in Industrial settings. In: Proceeding of the 15th EICAR Annual Conference, Hambourg (2006)
14. Ning, P., Cui, Y., Reeves, D.S.: Constructing Attack Scenarios through Correlation of Intrusion Alerts. In: Proceedings of the ACM Conference on Computer and Communications Security, Washington, D.C, November 2002, pp. 245–254 (2002)
15. http://www.modbus.org/

Towards Early Warning Systems – Challenges, Technologies and Architecture*

Martin Apel[1], Joachim Biskup[1], Ulrich Flegel[2], and Michael Meier[1]

[1] TU Dortmund, Computer Science VI, 44221 Dortmund, Germany
{martin.apel,joachim.biskup,michael.meier}@cs.tu-dortmund.de
[2] SAP Research, Vincenz-Prießnitz-Str. 1, 76131 Karlsruhe, Germany
ulrich.flegel@sap.com

Abstract. We present the architecture of an *automatic* early warning system (EWS) that aims at providing predictions and advice regarding security threats in information and communication technology without incorporation of cognitive abilities of humans and forms the basis for drawing a situation picture. Our EWS particularly targets the growing malware threat and shall achieve the required capabilities by combining malware collectors, malware analysis systems, malware behavior clustering, signature generation and distribution and malware/misuse detection system into an integrated process chain. The quality and timeliness of the results delivered by the EWS are influenced by the number and location of participating partners that share information on security incidents. In order to enable such a cooperation and an effective deployment of the EWS, interests and confidentiality requirements of the parties involved need to be carefully examined. We discuss technical details of the EWS components, evaluate alternatives and examine the interests of all parties involved in the anticipated deployment scenario.

Keywords: automated process chain, early warning, clustering, confidentiality, intrusion detection, signature learning.

1 Introduction

Along with the growing dependency of our society on information technology (IT) systems, concerns regarding IT security are becoming more urgent. While up to now primarily preventive measures and mechanisms have been focused, it becomes increasingly apparent that IT security cannot be achieved by prevention alone. Rather, preventive measures and reactive aspects need to complement one another. Precondition to reaction on security incidents is a dependable and timely detection of respective situations. A cooperative information exchange between different institutions is not only advantageous, but also mandatory in order to detect distributed and coordinated attacks. From a large-scale acquisition of pertinent information by an early warning system (EWS) arises the opportunity to draw up the situation picture that allows the detection of trends and upcoming threats, thereby allowing to take appropriate measures.

* This work was accomplished in cooperation with the German Federal Office for Information Security (BSI) and the German Federal Ministry of Economics and Technology (BMWi).

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 151–164, 2010.

The need for integrating data in order to construct such a situation picture is widely accepted (cf. e.g., [1,2,3,4,5]. However, typically there exist reservations concerning the distribution of information allowing outsiders insights into security incidents of individual institutions. These reservations are opposing the integration of information and so far prohibit the creation of a situation picture. A practical EWS needs to take the conflicting interests of the participating parties into consideration. A resolution of the conflicts can be achieved by using information reductions, e.g., pseudonymization.

Though the term EWS has been used in the literature, there is no common, accepted definition of what early warning is (and no differentiation to, e.g., intrusion detection systems (IDS)). A vague definition of an early warning information system (EWIS) is given by [6], who defines EWIS by sketching its purpose: 'EWIS assists experts and policy makers in assessing desired options for...' several particular security measures. By outlining one particular realization technique, [6] defines '[an EWS] for IT security surveillance is based on a specific procedure to detect as early as possible any departure from usual or normal observed frequency of phenomena.' A more general operational definition of early warning is used by [1]: 'In case of perceptible indicators, and no or (still) a low number of victims, information must be distributed, to help others – not yet victims, including response organizations, in order to avoid a major crisis.' We adopt the more declarative definition given by [7]: 'EWS aim at detecting unclassified but potentially harmful system behavior based on preliminary indications and are complementary to intrusion detection systems. Both kinds of systems try to detect, identify and react before possible damage occurs and contribute to an integrated and aggregated situation report (big picture). A particular emphasis of EWS is to establish hypotheses and predictions as well as to generate advices in still not completely understood situations. Thus the term early has two meanings, a) to start early in time aiming to avoid/minimize damage, and b) to process uncertain and incomplete information.'

This paper sketches the architecture of such an early warning system that is particularly targeting the malware threat and is currently under development. The fundamental requirements of an early warning system are a) automatic detection of known as well as unknown automated security violations, and b) automated indication of security incidents in the form of alerts, which can be combined into a situation picture. These capabilities shall be achieved by combining the following technologies to an integrated automated process chain: 1) capturing active malware using honeypot technology based malware collectors, 2) analysis of malware and generation of patterns for detecting malware (signatures) using machine learning techniques, 3) central consolidation and storage of the generated signatures, 4) distribution and deployment of signatures to signature-based detection systems, 5) central alerting to an emergency response center.

Our contribution is threefold. Firstly, we describe the architecture of our EWS in section 2, followed by a discussion of the anticipated deployment scenario in section 3. Secondly, we discuss the technical and organizational challenges, which need to be considered when implementing the EWS, and sketch technical details of the components of our EWS in section 4. Thirdly, deployment issues, in particular regarding participating parties, their interests and the resulting privacy and confidentiality requirements for an EWS, are examined in section 4.5. Finally, we discuss related work and summarize in section 5.

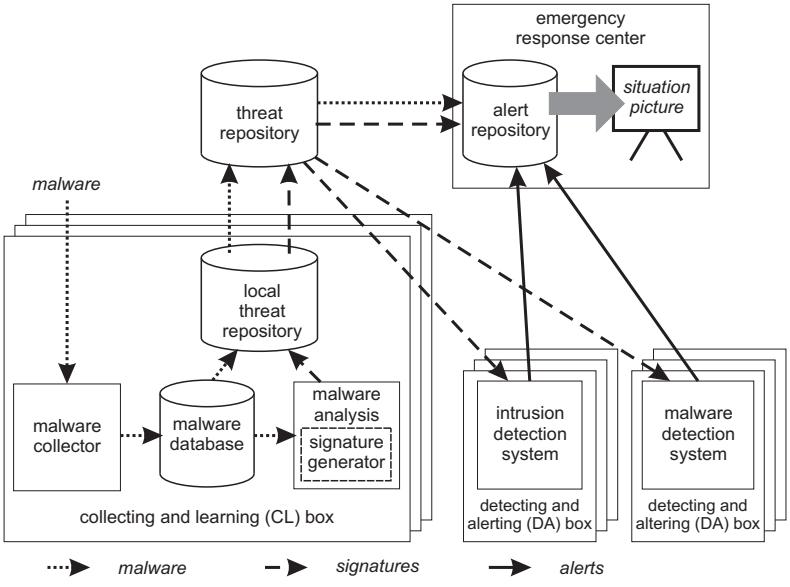


Fig. 1. Architecture of the early warning system

2 Architecture

We propose the following architecture for an EWS as shown in fig. 1. It comprises four basic components. The *collecting and learning (CL) box* bundles the functional components to collect malware, to analyze malware to extract features used for learning and to generate appropriate detection patterns. Automated signature generation is a primary focus of the described project. Because of the honeypot component placed upstream our approach can exploit the assumptions that the collected and analyzed files are malware indeed. The *threat repository* is used to centrally manage information on malware and detection criteria delivered by CL boxes. It supplies the information needed for detecting malware to the *detecting and alerting (DA) boxes*, which contain the functional components to detect respective security violations and to generate alerts. Alerts generated by the DA boxes as well as information on malware supplied by the threat repository form the basis for constructing a *situation picture* and they are centrally managed in the *alert repository*. Please note that, even if not the case in the particular design discussed here, multiple threat and alert repositories are possible and cooperative exchange of information between different EWS at the repository level can be foreseen. The overall procedure allows to match signatures for detecting misuse of observed threats in a timely and fully automated manner. It is designed to simultaneously achieve the following: retaining the advantage of misuse detection to provide specific alerts with a low false positive rate and compensating for the original weakness of misuse detection to detect only a priori known security violations.

3 Deployment Scenario – Parties and Interests

Both the protection level achieved for the participating systems and the quality of the situation picture depend on the number and placement of deployed CL boxes as well as DA boxes. The more CL boxes are deployed at suitable locations in the network the higher will be the likelihood that new malware is collected during an early stage of its distribution and signatures for detection systems are supplied by the threat repository early enough to observe, detect and restrict further distribution of the malware. The more DA boxes are suitably placed in the network the more comprehensive the information base for constructing the situation picture will be. The installation of a larger number of CL boxes as well as DA boxes in different network domains is anticipated as sketched in Fig. 2. A primary prerequisite is that the domain owners agree to cooperatively exchange information on security incidents. Without additional protective measures, there exists, e.g., for the owner of the threat repository the possibility to gain knowledge about the occurrence of security incidents in the domains of CL boxes. Analogously, the owner of the emergency response center gains insight into security incidents detected in the domains of DA boxes. Since this may conflict with the interests of the domain owners, an agreement about exchanging the required information is hard to achieve. To resolve this problem and to rebut respective reservations, we investigate requirements on data and privacy protection for required information flows and consider complying information reductions, e.g., pseudonymization (cf. 'data and confidentially protection' in fig. 2). In particular, such information reductions effectively confine owners of central components to acquire sensitive information, without affecting the functionality of the system or significantly reducing the quality of the situation picture. In the following we refine the deployment scenario (cf. fig. 2) by discussing the suitable parties for operating the system components and elaborating on their interests.

For the current setting, we expect that the *alert repository* is operated by a government agency, such as the German Federal Office for Information Security (BSI), which

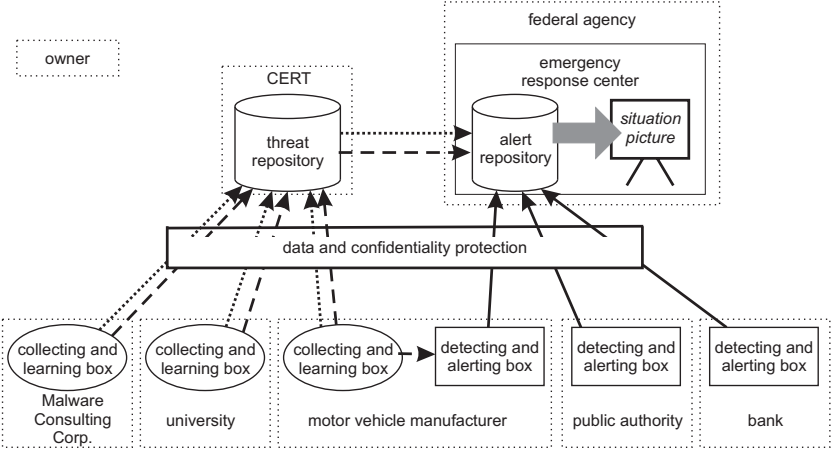


Fig. 2. Domains of an example deployment scenario

is officially commissioned to provide a situation picture. The agency might also outsource the operation to a CERT (computer emergency response team) or a private sector enterprise. Alternative models would be a private company providing the situation picture as a payable service, or a closed consortium of the EWS member organizations.

The *threat repository* may also be operated by a government agency, a CERT, a private sector company or a consortium of EWS member organizations. If the operating organizations of the alert and the threat repository differ, suitable combinations of operating organizations are subject to further investigation. Organizations that already operate an IDS are natural candidates for member organizations of an EWS. The primary driving force behind installing a *DA box* is the organization's interest in the information provided by the situation picture and its value for safeguarding its assets. All sectors that rely on IT services are promising candidates for operating DA boxes, i.e., government agencies, academia and private sector organizations.

For the *CL boxes*, the same consideration as before applies. In particular, a service provider may be specialized on providing a broad and representative collection of malware samples. While it is perfectly possible that an organization specializes either on collecting malware and generating signatures, or on detecting incidents and providing alerts, a combined operation of DA and of CL boxes is valuable. Operating both kinds of boxes allows the organization to generate new signatures to cater to its own need for detecting locally occurring malware and thwarting its recurring occurrence.

4 Details

4.1 Malware Collector

For collecting malware, server-honeypots like Nepenthes [8] and Amun [9], which provide vulnerable services in order to attract spreading malware, are used in our EWS. These systems are presently able to collect malware that exploits known vulnerabilities. Therefore they emulate vulnerable services. Spreading malware exploits vulnerable services to enter the victim system and downloads the malware binary from a *malware distribution system* (MDS) afterwards. Payload targeting an emulated service vulnerability is captured and analyzed by the malware collector, the URL of the malware binary on the MDS is extracted and the malware binary is downloaded eventually.

4.2 Malware Analysis System – Sandbox

Malware analysis systems are used for inspecting and extracting features of malware that are appropriate for characterizing and distinguishing malware and benign programs. Static and dynamic analysis can be distinguished. *Static analysis* focuses on static features that can be directly extracted from a malware sample. Sequences of data and instructions of a sample are typical examples. Morphing (aka obfuscation) techniques and tools demonstrate that programs of similar functionality do not need to share similar instruction and data sequences, leading to polymorphic and metamorphic variants of malware [10]. Consequently, higher-level structural features like the control-flow graphs (CFG) of programs, which can be extracted by disassemblers, are

studied (e.g. [11]). These techniques assume that similar programs need to share a similar structure. Unfortunately, albeit not yet commonly used by today's malware authors, techniques exist to change the program structure without changing its functionality.

Dynamic analysis avoids these drawbacks by focusing on the dynamic behavior of a program, which can be extracted by executing a program and closely observing its activities, e.g., at the system call level. As polymorphic variants of the same malware are behaving almost identical, polymorphic samples are easily detected based on their common behavior [12]. Even some metamorphic techniques can be thwarted that way. A difficulty with dynamic techniques is to trigger the malicious execution path of the program under observation. Nevertheless, we postulate that dynamic analysis is the most promising alternative for malware analysis and detection. The actions of a malware are described by its interaction with the operating system (OS), specifically the system calls it uses. The *trace of a malware* represents the list of all system calls that are performed by the malware. As a malware can start multiple threads or even processes, the trace contains multiple blocks, each one describing one thread. Inside of the blocks the system calls are ordered chronologically. In the context of the project described here, CWSandbox [13] is used as dynamic malware analysis system.

4.3 Automatic Signature Creation

The creation of behavioral signatures consists of two steps (fig. 3). The first step is to group similar behavior reports together. To do so, a function is needed, that computes the (dis)similarity of two traces. Which function is used has a great impact on the resulting groups, as it determines the used features and their weighting. In the second step a signature is created for each of the resulting groups. During this step an additional set of traces is used, which contains the behavior of benign executables and is called the *good pool*. The usage of the good pool helps keeping the *false-positive* rate low.

Clustering. The grouping of malware traces can be done using a cluster algorithm (which groups elements from a set S regarding certain features into subsets C_i (clusters) such that $S = \bigcup_i C_i$). Most algorithms will create disjoint clusters, i.e., $C_i \cap C_j = \emptyset$ for $i \neq j$. Unfortunately, there exist malware samples that show behavior common to multiple families, and thus their traces could be put into different groups. There are even samples that contain more than one individual malware. Overlapping cluster algorithms that

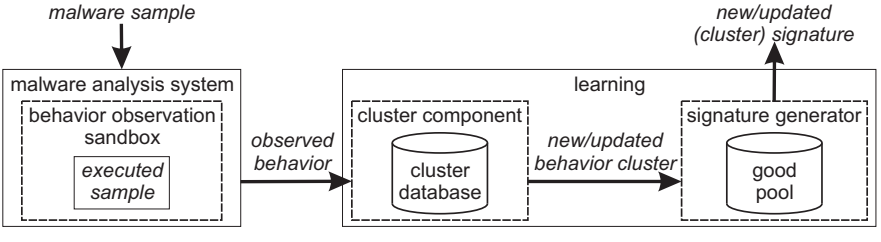


Fig. 3. Signature generation process

allow the presence of one element in multiple C_i are therefore of special interest to us. A cluster algorithm suited for our needs should fulfill the following criteria:

1. The only input parameter the algorithm needs are pairwise distances. This means that no special metric is required and makes the used metric interchangeable.
2. It should be able to handle noisy data (be robust). This provides the algorithm with a better ability to handle outliers and enables stronger generalization.
3. The number of clusters to be generated should not be part of the input, as an appropriate choice depends on the data set and varies between data sets.

Cluster algorithms fulfilling the above criteria are for example hierarchical clustering algorithms (single-, complete-link, WPGMA, UPGMA) [14] and fuzzy clustering [15] which also provides overlapping clusters, which would certainly improve the quality of the clustering, though it is not necessarily needed. The quality of the clustering depends on the *distance metric* used. We have to compute distances between traces of every pair of malware samples, and to do so repeatedly and thus highly efficiently. The distance between two traces relies on the properties that the metric will take into account. These properties can be quite different between different metrics and it has to be carefully examined which one to choose.

A well known approach is to treat the traces as long strings and use the *edit distance*, which computes the *cost* of making the two strings equal and relates these costs to the length of the strings. The edit distance has been used in malware analysis by Lee and Mody [16]. Unfortunately, the runtime complexity of the edit distance is $O(n^2)$ in the length of the strings. Another possibility is the use of the *normalized compression distance (NCD)* [17], which approximates the Kolmogorov complexity using compression algorithms. The runtime of *NCD* is tied to the runtime complexity of the used compression algorithm. It has been employed succesfully for malware analysis [18] and in many other fields. The used compression algorithm has to be choosen carefully as shown by Cebrián et. al. [19]. A fundamental problem with *NCD* is that it only measures the structural complexity of a string and ignores any semantic context. It is debatable whether the structural differences of the traces of different malware samples reflect their behavioral distance indeed. A quite different idea is based on an *embedding* function which embeds sequential data into a high-dimensional vector space using a formal language L. Each dimension of a vector corresponds to a specific word of L and holds the number of occurrences in the data. The dimensionality of each vector depends on the number of words in L which is usually very high. As the resulting vectors are very sparse, special data structures can be used to store them [20]. To compute the (dis)similarity between elements of the vector space the L_m (Minkowski) metrics can be chosen, which is defined as $L_m(x,y) = (\sum_{i=1}^{n} |x_i - y_i|^m)^{\frac{1}{m}}$. Another possibility is the usage of similarity coefficients.

Our experiments [21] showed that the usage of the Manhattan distance L_1 (along with tries) outperformed the NCD (using ppmd and lzma compression) as well as the edit distance. The edit distance yielded better results than the NCD variants. Thus, for further development of our EWS, we chose the vectorization using the Manhattan distance.

Behavioral Signature Creation. The creation of the signatures is performed for a group of malware traces found to be similar using the clustering approach. The goal

is to determine sequences of system calls that are shared among these traces, but are absent in *normal* programs taken from the *good pool*. One possibility is to find all substrings $s_1, ..., s_l$ shared between all traces in the group and build a signature that matches a sequence if all of the shared strings are contained. The shared substrings can be found using *generalized suffix trees*, which can be build in time $O(n_1 + n_2 + ... + n_m)$, where n_i is the length of the i-th trace, using the algorithm of Ukkonen [22]. Another approach relies on the embedding into a vector space and the use of a support vector machine (SVM) (which finds an optimal hyperplane between two sets of vectors). Choosing the sets S_m and S_b as the sets of vectors belonging to the malicious traces and the benign traces, respectively, the hyperplane found by a SVM can be taken as a signature.

Optimization. The clustering as well as the signature creation can be time consuming and thus, for the early warning context, demands to deploy optimization techniques like the following one. If a new sample arrives its behavior is tested against the existing signatures first. If any signature matches no further action is taken. If no signature matches, the distances of the new sample (resp. its behavior) to the existing groups of malware traces are evaluated. The new sample is added into the group that exhibits the smallest distance to the sample's trace and the signature for that group is recreated. As this procedure might worsen the quality of the clusters (creating a new singleton cluster or splitting an old cluster may be better), a complete re-clustering is performed periodically for all traces.

Validation. There are essentially two problems that can occur. The signature can be too specific or to general. To ensure that the signature is not too specific, a technique called X-fold-Cross validation is used. The group of malware traces for which a signature is to be created is divided into X parts. $X-1$ of these parts are used to create the signature while the remaining part is used for testing. If the signature is good and generalizes to a suitable extent, the traces not considered for signature creation should be detected as well. To ensure that the signature is not too general, the signature is tested against the *good pool*. X-fold-Cross validation can be used here as well.

4.4 Malware/Misuse Detection Systems

Misuse detection systems are used to detect security incidents based on observations of security relevant events and signatures. Examples of available solutions that support this functionality include intrusion detection systems, virus scanners and firewalls. It is beneficial to integrate as many of the available detection products into the EWS as possible. Due to space restrictions we abstain from discussing common technical requirements of detection systems here and focus on requirements that are specific to the deployment of the detection systems within the EWS.

When integrated with EWS, detection systems need to be able to receive new signatures from a threat repository, to deploy them on the fly, and to forward generated alerts to an alert repository. Further, the signatures supplied by the EWS need to be compatible with the employed detection system. In particular the feature domain used for generating signatures in the EWS and the features observed and analyzed by the detection systems need to compatible. That is, the behavioral features of malware that are extracted using CWSandbox and used for signature generation need to be observed/monitored by

the detection system in order to support effective matching of the supplied signatures. For integrating existing detection systems possibilities to transform signatures between different domains need to be realized.

Besides the integration of existing detection systems in DA boxes, a detection system called *jSAM (Java Signature Analysis Module)* is in under development that provides full support of the behavioral features used in EWS supplied signatures. Further highlights of jSAM are the expressiveness of the used signature language *EDL (Event Description Language)* [23], which is also used as signature transfer language inside the EWS, as well as the optimized matching strategies.

4.5 Privacy and Confidentiality Requirements

We first consider entities and their potential relation to personal data. Malware samples may contain (hidden) clues about their authors and endpoint addresses of *malware target systems* (MTS), which are targeted victims of attacks executed by malware (e.g., a denial of service attack) and *malware drop zones* (MDZ), where data copied from *victim systems* (VS) is uploaded. While MDZ may be operated by malware authors, this is usually not the case. The copied data is rather delivered to VS that are controlled by malware authors. VS often are poorly managed home computers operated by natural persons. Hence, we consider MTS as well as MDZ endpoint data as personal data. Malware spreads from so-called *malware host systems* (MHS), and the CL Box can observe the MHS endpoint of a malware trying to spread to the CL Box. From the observed exploit payload the CL Box can extract the *malware distribution system* (MDS) endpoint. In the vast majority of cases the MHS and MDS are VS. Hence, we consider MHS and MDS endpoint data as personal data. We do not protect the interests of malware authors and will therefore ignore them in the following.

Next, we inspect the data flow from malware VS via a CL box to the central threat repository (cf. fig. 4). The CL Box initially receives the exploit of a given malware and can identify the originating MHS. In a second step, the CL Box extracts the download endpoint of the (possibly different) MDS (cf. section 4.1) where the complete malware sample is situated and finally downloads it. As follows, the CL Box receives the following data concerning the attacking MHS: sending endpoint (IP, port) and payload (exploit); and from the malware sample serving MDS: download endpoint (IP, port, protocol-specific path) and payload (malware sample). In addition the following data can be determined: receiving endpoint (CL Box IP, port), name of the vulnerable service, timestamp, malware sample id (e.g., md5 hash value). For further processing, the CL Box persists the following data to the local threat repository, which then will be sent to the global threat repository (as defined above): sending endpoint, receiving endpoint, download endpoint, name of vulnerable service, malware sample, timestamp, malware sample id. In addition malware analysis systems may extract additional potentially confidential data from malware samples (e.g., endpoint data of MTS, MDS or MDZ) and provide it to the threat repositories. CL boxes also provide automatically generated signatures to the threat repository, where a signature detects an observable behavioral pattern of the collected malware sample. The signatures are distributed to the DA boxes for malware detection. DA boxes may use a network interface (*observing endpoint*) for observing network behavior. When a DA box successfully matches a signature to observed behavior

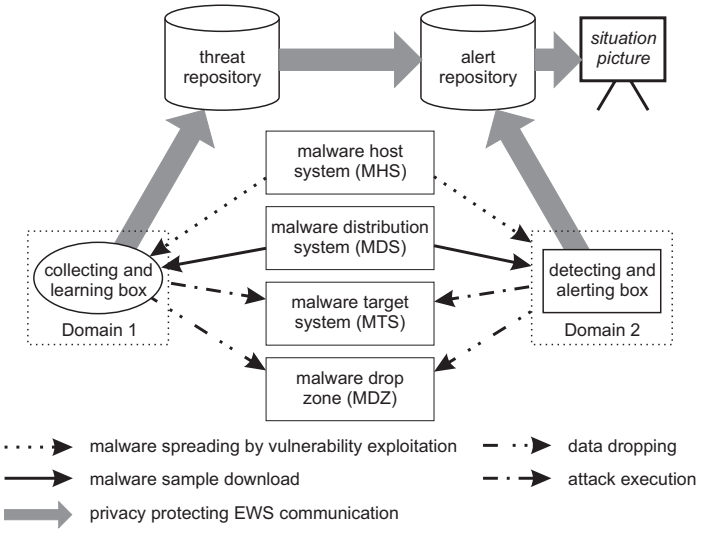


Fig. 4. Flow of private and confidential data

it generates an *alert* that primarily contains a timestamp and the name of the matched signature.

Additionally, we classify the functionality of the EWS into two broad classes: analysis and reaction. Analyzing the features delivered by the CL and DA boxes means relating events (malware samples, alerts) by means of their features. This requires that the features that are used to relate events to each other be *linkable* (*linkability* requirement). Note that features can be transformed, such that they are still linkable, but do not easily give away the original information [24]. The EWS warns its participants by providing names or port numbers of services that are currently critical, endpoints from where malware is downloaded, notifies VS endpoints that have been infected by malware, etc. To be able to do so and to make the distributed information actionable for the receivers, the information can be provided in transformed form, e.g., encrypted, but the receivers must be able to *disclose* the original data (*disclosure* requirement). As an example, a DA box may use the MDS address feature to block outbound malware downloads at the firewall of the domain. For this the DA box must be able to see the IP address of the MDS in the clear.

We finally examine the interests of the involved parties regarding the ability to link and disclose personal or otherwise confidential data from the data flows considered above. Malware VS are usually involved when outbound and inbound activity of MHS is recorded (exploit phase), when a malware sample is downloaded from MDS, when malware attacks MTS, as well as when malware copies data from VS to MDZ. Addresses of MDS, MTS and MDZ may be hardcoded in the malware sample. Since we consider VS to be identified with the operating persons, the related data is personal data. These persons are assumed to be interested in protecting their privacy and as such are not interested in the disclosure and linkability of their personal data. They however are interested in the disclosure and linkability of the personal data of other VS attacking them, to be able to defend their assets or claim compensation.

Table 1. Summary of feature specific interests of the involved entities and interest support by the proposed balance of conflicting interests – c: confidentiality of own/domain-local feature; l: linkability of remote feature; d: disclosure of remote feature; blank: party has no interest; I: party's interest is not supported; S: party's interest is fully supported; P: party's interest is supported merely against selected other parties

feature	controlled by attacker	victim			CL box	DA box		threat repository		alert repository	
		c	l	d	c	l	d	l	d	l	d
timestamp	Yes				P	S	I			S	S
alert signature name	Yes				P	S	I		S	S	S
sending endpoint of MHS	Yes	I	I	I	P	S	I		S	S	S
receiving endpoint of MTS	Yes	I	I	I	P	S	I		S	S	S
download endpoint of MDS	Yes	P	I	I	I	S	S		S	S	S
upload endpoint of MDZ	Yes	P	I	I	I	S	S		S	S	S
vulnerability module name	Yes				I	S	S		S	S	S
receiving endpoint of CL box					S	I	I		S	S	
observing endpoint of DA box					S	I	I		S	S	
malware exploit payload	Yes				I	S	S		S	S	
malware sample payload	Yes	P	I	I	I	S	S		S	S	

CL boxes, as well as DA boxes are assumed to be operated by organizations, not by natural persons. We also assume that the boxes are provided as stand-alone systems that have no users, except for initial administrative purposes. As a result we do not need to consider personal data regarding these boxes. It remains to consider the remaining data that is sent out to the repositories. This data can be considered sensitive when observed by customers or competitors, such that the organization is interested in keeping them confidential. At the same time a given organization may be interested in the malware incident data of other organizations to gain a competitive edge.

The threat repository and the alert repository merely collect data from the CL boxes and DA boxes, respectively. The data in the threat repository may be refined and enriched manually, but it will still not refer to the operating organization. Hence, we do not need to consider personal data of the operating organizations here. The role of the global threat repository is providing information to DA boxes for detecting and possibly directly blocking malware. For the purpose of transitory blacklisting sites involved in the outbreak of a given malware, the threat repository needs to disclose the endpoints of MDS, as well as MDZ. Proactively patching or shutting down services would be enabled by disclosing the receiving port numbers of the malware collector. Repository data refinement and analysis for enrichment is enabled by disclosure of the payloads (exploits and malware sample). Duplicate elimination is necessary for efficiency reasons. The alert repository needs to be able to link all data items to create a situation picture. Some data needs to be disclosed for detailed reporting and advisories for time-critical and transitory blacklisting.

Table 1 summarizes the interests of all parties, which, obviously, may conflict. For example, VS want to keep their endpoints confidential and CL boxes also want to keep their existence confidential. However, e.g., DA boxes of other domains are generally

interested in disclosing these features. It is therefore necessary to define a suitable balance between the conflicting interests. In some cases it suffices to support a given interest in linkability or disclosure only for the repository owners, in other cases it is necessary to support the given interest also for DA boxes. The proposed balance supports the confidentiality requirements of VS only partially: box owners and repository owners can in most cases link and disclose the VS endpoint. The confidentiality requirements of CL and DA box owners can only be supported for the receiving CL endpoint, the observing DA endpoint and the timestamp feature; for the other features it is necessary to let other DA boxes link and disclose them to allow for timely response. An array of techniques to balance interests was discussed in [25].

5 Related Work and Summary

A few EWS has been proposed in the literature. They all have the sharing and central collection of information in common but they can be differentiated regarding the kind of information that is processed and correlated. Another differentiating feature of EWS is the way they establish hypotheses and predictions and generate advice. Existing systems support information aggregation and visualization as well as statistic analysis but predictions or advice generation is left to the human user. In the following we discuss and compare related approaches regarding these differentiating features.

DShield's Internet Storm Center [2] operates on firewall logs of several organizations and incorporates human interpretation and action in order to generate predictions and advice. MyNetWatchman [26] also processes firewall logs of multiple organizations but supports automatic generation of email notifications. eCSIRT.net system [3] comprises of a sensor network of intrusion detection systems. It collects and correlates alerts of these system which are visualized for human inspection. The Internet Motion Sensor (IMS) [27] statistically analyses dark net traffic that need to be interpreted by humans. SURFids [28] stores malware binaries collected by a network of Nepenthes sensors in a central database and supports generation of several statistics. The Deepsight system uses about 19000 sensors that provide IDS alerts, firewall logs and honeynet data. Human analysis and data mining is incorporated in order to provide statistics. Zou et al [29] propose the Malware Warning Center that realizes worm detection based on an epidemic propagation model. It is focused on worms that uniformly scan the internet and aims at early detection of worm epidemics. The Internet Malware Analysis System (InMAS) [4] collects malware using honeypots, honeyclients and spamtraps and analyses collected files using CWSandbox. Predefined statistics on collected information can be visualized using a statistic backend. The Internet Analysis System (IAS) [30] collects and visualizes statistics on network packet data in order to support detection of anomalies by humans. eDare (Early Detection, Alert and Response system) [31] and the Agent-based EWS [5] are two further proposals to support early warning. While Agent-EWS basically propose to centrally collect information of different sensors the eDare systems propose to use supervised machine learning techniques for detecting unknown malware. ADWICE [32], which is part of the Safeguard project, uses unsupervised learning to generate a model of normal traffic and performs outlier detection to find anomalies. The CarmentiS project [1], advanced by the German CERT association, is

another initiative towards cooperative sharing, central storage and visualization of different kinds of sensor data, which is planed to be extended to by correlation techniques in order to automatically generate advice and predictions.

In comparison to these approaches, our system operates on different kinds of information, which are potentially new malware samples (collected, e.g., by Nepenthes), malware behavior (extracted, e.g., by CWSandbox), automatically generated and distributed signatures (generated using machine learning techniques) as well as detection alerts (generated by detection systems using the signatures). Due to the automatic derivation of new signatures and central reporting of occurrences of new malware threats, it forms a basis to generating predictions and advice without incorporation of cognitive abilities of humans and is therefore a large step towards an *Automatic EWS* that automatically contribute to a situation picture.

Accordingly, in this paper we have described the architecture of our automatic EWS and discussed details regarding alternative implementations of its key components. While malware collection and analysis is mainly realized using existing approaches, clustering of malware behavior and generating behavior signatures in the context of our EWS are focus of our research. Enforcing a balance between conflicting confidentiality and availability requirements is another key research challenge of this ongoing project. Based on a discussion of a deployment scenario, interests and requirements of all involved parties are examined. A balance between conflicting interests is proposed and mechanisms appropriate to enforce this balance are mentioned. Besides completing ongoing efforts to implement the EWS components, future work in this project includes the deployment and evaluation of the EWS.

References

1. Grobauer, B., Mehlau, J., Sander, J.: Carmentis: A co-operative approach towards situation awareness and early warning for the internet. In: Proc. of IMF 2006. LNI, vol. 97, pp. 55–66. GI (2006)
2. DShield: DShield website (2008), http://www.dshield.org
3. Network, T.E.C.: The European CSIRT Network Website (2008), http://www.ecsirt.net/
4. Engelberth, M., Freiling, F., Göbel, J., Gorecki, C., Holz, T., Trinius, P., Willems, C.: Frühe Warnung durch Beobachten und Verfolgen von bösartiger Software im Deutschen Internet: Das Internet-Malware-Analyse-System (IAS) (in German). In: Sichere Wege in der vernetzten Welt – Tagungsband zum 11. Deutscher IT-Sicherheitskongress (in German), pp. 353–367. SecuMedia Verlag (2009)
5. Bsufka, K., Kroll-Peters, O., Albayrak, S.: Intelligent network-based early warning systems. In: López, J. (ed.) CRITIS 2006. LNCS, vol. 4347, pp. 103–111. Springer, Heidelberg (2006)
6. Gattiker, U.: The Information Security Dictionary. Kluwer, Dordrecht (2004)
7. Biskup, J., Hämmerli, B.M., Meier, M., Schmerl, S., Tölle, J., Vogel, M.: 08102 working group – early warning systems. In: Perspectives Workshop: Network Attack Detection and Defense. Dagstuhl Seminar Proceedings, vol. 08102 (2008)
8. Baecher, P., Koetter, M., Holz, T., Dornseif, M., Freiling, F.: The Nepenthes platform: An efficient approach to collect malware. In: Zamboni, D., Krügel, C. (eds.) RAID 2006. LNCS, vol. 4219, pp. 165–184. Springer, Heidelberg (2006)
9. Amun: Python Honeypot, http://amunhoney.sourceforge.net/
10. Aycock, J.: Computer Viruses and Malware. Springer, Heidelberg (2006)

11. Dullien, T., Rolles, R.: Graph-based comparison of executable objects. In: Proc. of SSTIC 2005 (2005)
12. Rieck, K., Holz, T., Willems, C., Düssel, P., Laskov, P.: Learning and classification of malware behavior. In: Zamboni, D. (ed.) DIMVA 2008. LNCS, vol. 5137, pp. 108–125. Springer, Heidelberg (2008)
13. Willems, C., Holz, T., Freiling, F.: Toward automated dynamic malware analysis using CWSandbox. IEEE Security & Privacy 5(2), 32–39 (2007)
14. Lance, G.N., Williams, W.T.: A general theory of classificatory sorting strategies: II. Clustering systems. The Computer Journal 10(3), 271–277 (1967)
15. Frigui, H., Krishnapuram, R.: A robust clustering algorithm based on competitive agglomeration and soft rejection of outliers. In: Proc. of Computer Vision and Pattern Recognition, vol. 550. IEEE, Los Alamitos (1996)
16. Lee, T., Mody, J.: Behavioral classification. In: Proc. of EICAR 2006 (2006)
17. Cilibrasi, R., Vitanyi, P.: Clustering by compression. IEEE Trans. on Information Theory 51, 1523–1545 (2005)
18. Bailey, M., Oberheide, J., Andersen, J., Mao, Z., Jahanian, F., Nazario, J.: Automated classification and analysis of internet malware. In: Kruegel, C., Lippmann, R., Clark, A. (eds.) RAID 2007. LNCS, vol. 4637, pp. 178–197. Springer, Heidelberg (2007)
19. Cebrian, M., Alfonseca, M., Ortega, A.: Common pitfalls using the normalized compression distance. Comm. in Information and Systems 5(4), 367–384 (2005)
20. Rieck, K., Laskov, P.: Linear-time computation of similarity measures for sequential data. Journal of Machine Learning Research 9, 23–48 (2008)
21. Apel, M., Bockermann, C., Meier, M.: Measuring similarity of malware behavior. In: Proc. of 34th LCN 2009. IEEE Computer Society Press, Los Alamitos (2009)
22. Ukkonen, E.: On-line construction of suffix trees. Algorithmica 14(3), 249–260 (1995)
23. Meier, M., Schmerl, S., Koenig, H.: Improving the Efficiency of Misuse Detection. In: Julisch, K., Krügel, C. (eds.) DIMVA 2005. LNCS, vol. 3548, pp. 188–205. Springer, Heidelberg (2005)
24. Flegel, U., Biskup, J.: Requirements of information reductions for cooperating intrusion detection agents. In: Müller, G. (ed.) ETRICS 2006. LNCS, vol. 3995, pp. 466–480. Springer, Heidelberg (2006)
25. Flegel, U.: Privacy-Respecting Intrusion Detection. Springer, Heidelberg (2007)
26. MyNetWatchman: MyNetWatchman website (2008), http://www.mynetwatchman.com
27. Bailey, M., Cooke, E., Jahanian, F., Nazario, J., Watson, D.: The internet motion sensor - a distributed blackhole monitoring system. In: Proc. of NDSS 2005, The Internet Society, pp. 167–179 (2005)
28. SURFids: SURFids Development Homepage (2008), http://ids.surfnet.nl
29. Zou, C., Gao, L., Gong, W., Towsley, D.: Monitoring and early warning for internet worms. In: Proc. of ACM CCS 2003, pp. 190–199 (2003)
30. Waibel, F.: Das Internet-Analyse-System (IAS) als Komponente einer IT-Sicherheitsarchitektur (in German). In: Sichere Wege in der vernetzten Welt – Tagungsband zum 11. Deutscher IT-Sicherheitskongress (in German), pp. 281–296. SecuMedia Verlag (2009)
31. Elovici, Y., Shabtai, A., Moskovitch, R., Tahan, G., Glezer, C.: Applying machine learning techniques for detection of malicious code in network traffic. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007. LNCS (LNAI), vol. 4667, pp. 44–50. Springer, Heidelberg (2007)
32. Burbeck, K., Nadjm-Therani, S.: Adwice – anomaly detection with real-time incremental clustering. In: Park, C.-s., Chee, S. (eds.) ICISC 2004. LNCS, vol. 3506, pp. 407–424. Springer, Heidelberg (2005)

CII Protection - Lessons for Developing Countries: South Africa as a Case Study

Mboneli Ndlangisa and Deon Herbst

ISCOTRA, 253 Silver Oak Ave, Waterkloof Ridge, Pretoria, South Africa
deon@dst-solutions.com

Abstract. We explore the process followed in formulating the South African CII (Critical Information Infrastructure) identification criteria and its application. We report on a three pronged approach that defines National Security, severity of CII security incidents and roles and responsibilities for CII protection. Our Criteria assumes the existence of basic ICT security roles within a country as per application of the South African criteria and its suitability for a country with limited resources. We conclude by recommending a CII protection approach that is best suited for developing countries based on our experiences.

Keywords: National Security, Critical Infrastructures, Critical Information Infrastructures, South African CII Identification Criteria.

1 Introduction

The digital revolution or the information age has come with many life-changing benefits and some of these benefits are reflected in the management of Critical Infrastructures (CI). Until recently, we did not have early flood warning systems, drought-monitoring infrastructures, sophisticated real-time electricity supply demand management systems and many others. At the core of all of these systems are the various ICT systems that support the management of CI and these systems are called Critical Information Infrastructures (CII).

There is no single globally agreed definition of Critical Infrastructures or of Critical Information Infrastructures and at times these two terms are used interchangeably. Critical Infrastructures are commonly understood as infrastructures or assets the incapacitation or destruction of which would have a negative impact on National Security and the economic and social welfare of a nation, Dunn, M. and Wigert, I. (2004). On the other hand, Critical Information Infrastructures are a component of the CI i.e. they are the ICT systems or cyber based systems crucial for the effective operation of the CI and they usually play a crucial role in connecting different CI, managing CI and making CI interrelated and interdependent.

The importance of CII and its role in cyber defense is acknowledged by the United Nations General Assembly (58/199) in as far as 2003 where it was noted amongst other things that:

- Recognizing the importance of ICTs in the promotion of socio-economic development and the provision of essential goods and services, the conduct

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 165–175, 2010.

of business and the exchange of information for Governments, businesses, other organizations and individual users, is essential.
- Recognizing that each country will determine its own critical information infrastructures.
- Noting that effective critical infrastructure protection includes, inter alia, identifying threats to, and reducing the vulnerability of, critical information infrastructures, minimizing damage and recovery time in the event of damage or attack, and identifying the cause of damage or the source of attack.

The above pronouncements from the UN General Assembly form the founding principles of our work i.e. the design and implementation of a CII identification criterion that is best suited for a developing country and South Africa is used as a case study. The model is designed to be generic, cost friendly and it applies the ITU principles from the "Generic National Framework for Critical Information Infrastructure Protection" Suter, M. (2007).

2 CII Protection: Developed versus Under-Developed Countries

A number of approaches on implementing CII protection programmes have been published and these run from the more mature programmes such as the US National Information Infrastructure Protection Plan, the Canadian programme – Office of Critical Infrastructure Protection and Emergency Preparedness (OCIPEP), Australia's Critical Infrastructure Group (CIPG) and many others from Western Europe and North America. Other countries have developed CII/CI programmes around the current existing structures but there is no specific agency looking after CII/CI protection. Most of the developing countries will share this structure where the CII/CI activities will be shared amongst a number of departments. For a more detailed analysis of the CII/CI protection refer to the CIIP Handbook Series from 2004, 2006, and 2008, published by the Swiss Federal Institute in Zurich.

Suter (2007) introduced a generic model for CII protection which is based on the Swiss experiences. The only limitation of this model is that it assumes the existence of quite a number of capabilities i.e. skills, National Computer Security Incident Response Team (CSIRT) and the political will, within the country.

3 CII Protection: The South African Situation

Looking at South Africa, there are three Acts relating to the protection of CII/CI.

They are:

- Electronic Communications Security Pty (Ltd) Act (Act 68 of 2002) – Section 21 refers to the protection of Critical Communications Infrastructures that are owned by the state.
- Electronic Communications and Transactions Act (Act 25 of 2002) – Chapter 9 of this Act refers to the identification of Critical Databases across South Africa.

- National Key Points Act 102 of 1980 – This act is the equivalent of the CI Protection Acts in other countries. It is based on the importance of the physical infrastructure to the South African Economy.

In summary, the three Acts fall within three separate national departments: Intelligence, Communications and Defense. There is no specific legislation covering the CII/CI protection. This is the case in most developing countries. The question then arises: "How does one identify and protect the CII while going through the iterative process of defining Parliamentary Policy Approval - the country's strategy, Private-Public Partnerships-collaboration framework, National Incident Management Capability (National CSIRTs), Curbing Cyber Crime, and Promoting a Cyber security awareness culture as prescribed by the ITU Draft ITU National Cyber security/CIIP Self-Assessment Tool(2008)?

South Africa, like many other developing countries, is vulnerable to Cyber Crime, Terrorism and Espionage, Foreign Intelligence infiltration and innocent CII system failures.

The challenge would be, given our limited financial and technical resources, what the best way would be to achieve what most First World countries have achieved over the last 10 years, in less than 2 years i.e. the strategy and legislative process can span up to 3 to 4 years whilst nothing could be done on the protection side.

It was decided to start three simultaneous processes:

1) Political Engagement to revise laws and produce a protection strategy
2) Identify all Critical Information Infrastructures and
3) Establish a National CSIRT.

The focus of this paper is on the South African CII Identification Criteria and its implementation.

4 Critical Infrastructure vs. Critical Information Infrastructure

As mentioned above, South Africa has the National Key Points Act of 1980 that identifies (on a continuous basis) a list of Critical Infrastructures (National Key Points) that are critical to National Security. The approach followed was to identify ICT systems within these structures that are critical for the minimum operations of these structures. In short, our model defines Critical Communications as certain/particular ICT Systems within the National Key Point that are critical for its minimum operation.

This approach is more cost-effective in the sense that it does not assume that if a place is a National Key Point, it must necessarily get the maximum security. This amounts to savings in time for Cyber attacks or failure readiness and resource usage.

5 The South African Approach

Our approach had the following steps:

- A comparative study on CII identification from a subset of First World economies (UK, USA), of developing countries (Singapore, India, Malaysia)

and International Organizations like NATO, EU , World Bank the United Nations and the International Telecommunications Union (ITU).
- The production of a South African CII Identification Criteria model – The process included testing the application of the criteria and CII object model.
- The identification of CII – this involved a risk assessment process – BIA, Asset Classification, Vulnerability assessment.

5.1 Comparative Analysis

Each of the chosen countries and organizations was studied more closely, in particular with regard to the following fields:

- Critical sectors identified within each country or group.
- Modeling methods, statistical approaches and information gathering techniques used.
- Policies – National Strategy on Cyber Security, CII, Cyber Crime or Terrorism
- Public agencies involved versus having one central agency looking after the overall CII protection.
- Public-Private partnerships - The level of engagement in dealing with CII in the private domain i.e. private CSIRT and in the National CSIRT.
- Early warning and public outreach methods.
- Law and legislation.
- Institutionalized vs. Legislated approach to CII protection.

The above focus areas build on the studies of Dunn, M. and Wigert, I. (2004) and the 2008 Draft ITU National Cyber Security/CIIP Self-Assessment Tool. The findings were that there is a need for a global regulatory framework on Cyber Crime and terrorism. As an example at the European level, the Council of Europe Convention on Cyber crime and the European Framework Decision on Attacks against Information Systems are currently among the most important pillars of transnational cyber-security legislation efforts. Another crucial point is that the developing countries evaluated did not have a central CII protection office (Malaysia was working on forming the International Multilateral Partnership Against Cyber Threats IMPACT).

5.2 South African CII Identification Criteria

In South Africa, CIIs are defined as all ICT systems, data systems, databases, networks (including people, buildings, facilities and processes), that are fundamental to the effective operation of the state. Due to the critical nature of these systems in the effective operation of the state, any extended unavailability, dysfunction, impairment or destruction of such infrastructures would have a negative impact on National Security.

Given the above definition, the next step is to determine/define the pillars of National Security. This would differ from country to country but the principle will remain the same. The criterion for the Identification, classification and protection is based on a three pronged approach, namely:

- The extended unavailability, impairment and dysfunction of a Communications Infrastructure must have an impact on any one or all of the four pillars namely National Law and Order, Social Services, Economy and Environmental matters.
- Once the impact has been determined, the severity is classified in terms of three severity levels such as 5 - maximum severity, 3 - moderate and 1 - minimum severity.
- The protection of the CCI is based on the impact of the extended unavailability and its severity on any one or all of the four pillars.

The pillars of national security represent the clusters of the South African Government and within these pillars are a number of government departments or sectors of the economy. This again can be easily adapted to fit any country. The clusters can be best summarized as follows:

- National law and Order - The CIIs are ICT systems that, if unavailable, dysfunctional, destroyed and/or compromised, may render the State institutions for the preservation of law and order ineffective. Some of the State institutions that may be affected in this regard would be Military, Judiciary, Policing and Intelligence.
- Social Services – The CIIs are ICT systems that, if unavailable, dysfunctional, destroyed and/or compromised, may render the State institutions for the preservation of social services such as education, payment of social grants, hospitals and clinics, Epidemic Disease Control, Sanitation and clean Water and Emergency services.
- Economy – The CIIs are ICT systems that, if unavailable, dysfunctional, destroyed and/or compromised, may have a negative impact on South African economy i.e. the effective operation of the industrial community (Agriculture, Mining, Telecommunications, Energy, and others); Financial Institutions, Labor.
- Environment – Communication systems that if compromised may affect effective functioning of Weather reporting services (Tsunami early warning systems), Water and Air pollution monitoring systems will all be deemed as CIIs.

Our approach differs from other CII identification methods in that it captures CII as an issue of national security, an issue of economic health, an issue of law enforcement and lastly an environment issue. This approach is more comprehensive in comparison to other countries as studied by Dunn, M. (2005).

In determining which ICT systems form part of CII, each system has to be measured against a research generated benchmark.

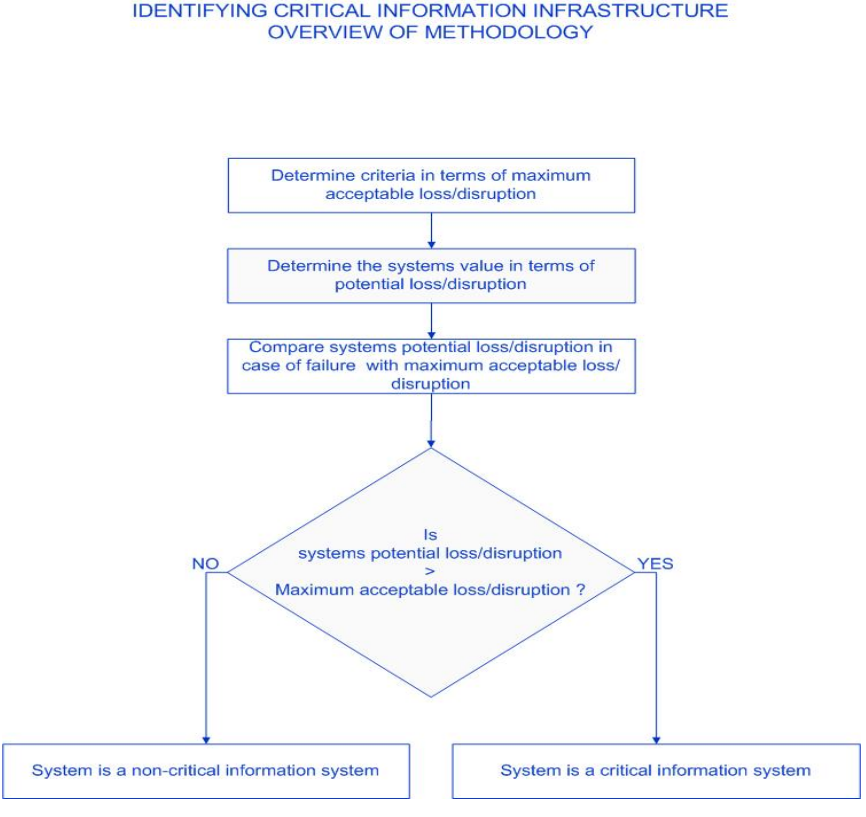
There are three main steps in this process: creating the benchmark, analyzing each system within each sector and then comparing its output against the created benchmark.

- Part 1 Create the benchmark score (Acceptable Levels of loss/disruption), the feedback of these answers needs to be normalized and preferably put in a matrix form or object modeled form for easy comparison with the results that are forthcoming from Part 2.

- Part 2 Analyze and determine the system's generated loss/disruption in case of failure, this is also a set of questions, but these questions are leveled at the system owners (managers responsible for communications infrastructure within each of the sectors/sub-sectors). The results of these questions are also put into a matrix or object modeled form similar to that of Part 1 for easy comparison.
- Part 3 Compare system failure generated loss with benchmark loss and categorize system as CII or not. This process entails the comparison of the analyses output matrix with the created benchmark output matrix to determine if that specific analyzed system, falls within CII or not.

The Acceptable Loss Indicator, Potential Loss Indicator will initially be biased towards the core functionality of the sector under investigation. These Loss Indicators will, however, be extended (i.e. vertical and horizontal interdependencies) during the process to indicate loss from a National Security point of view – that is, it will include loss within National Law and Order, Social Services, Economy and Environment.

The process of identifying CII can be best summarized by the following diagram:



One of the challenges of identifying CII is the ability to link interdependencies, or rather to have a CII object structure that caters for this. Rinaldi, S.M., Peerenboom J.P., Kelly T.K.(2001). This problem was highlighted by Dunn (2005) who emphasizes the need for a CII assessment tool-kit that caters for all levels of interdependencies.

Our CII object structure looks as follows:

5.3 CII Object Structure

SECTOR
DEPARTMENT
SYSTEM NAME
SYSTEM CHARACTERIZATION
MAXIMUM DOWN TIME ALLOWED
UNIQUE SYSTEM ID NO
NORMALISED POTENTIAL LOSS INDICATOR
CURRENT CRITICAL STATUS
ATTRIBUTES

- INFORMATION
    - Address
    - Location (GPS Coordinates)
    - Currently registered as National Key Point
    - Ownership
    - Current Custodian/Responsible Department
    - Usage/Function
- VERTICAL DEPENDENCIES (ICT Infrastructure)
    - Inward dependency SYSTEM ID 1
    - Type of link to SYSTEM ID 1
    - ……...
    - Outward dependency SYSTEM ID 1
    - Type of link to SYSTEM ID 1
    - …….
- VERTICAL DEPENDENCIES (Other Infrastructure)
    - Inward dependency SYSTEM ID 1
    - Type of link to SYSTEM ID 1
    - ……
    - Outward dependency SYSTEM ID 1
    - Type of link to SYSTEM ID 1
    - ……..
- HORIZONTAL DEPENDENCIES (ICT Infrastructure)
    - Inward dependency SYSTEM ID 1
    - Type of CII link to SYSTEM ID 1
    - Sector of other system
    - ……

- Outward dependency SYSTEM ID 1
- Type of CII link to SYSTEM ID 1
- Sector of other system
- ……

- HORIZONTAL DEPENDENCIES (Other Infrastructure)
  - Inward dependency SYSTEM ID 1
  - Type of CII link to SYSTEM ID 1
  - Sector of other system
  - ……..
  - Outward dependency SYSTEM ID
  - Type of CII link to SYSTEM ID 1
  - Sector of other system
  - ……
- SYSTEM THREATS
  - Threat 1
  - ……..
- SYSTEM VULNERABILITIES
  - Vulnerability 1
  - …….....
- CURRENT CONTROLS
  - Control 1
  - ……
- SYSTEM FAILURE CONSEQUENCES (IMPACT/HARM)
  - Duration of system disruption leading to critical situation
  - Number of people affected
  - Systems Affected (horizontally and vertically)
  - Related infrastructures affected (horizontally and vertically)
- LAST SECURITY AUDIT
  - Audited by
  - Date of audit

The above object structure strives to capture all the attributes of each CII and its relationships with other entities both ICT and non ICT i.e. National Key Points.

5.4 CII Object Structure – Cost/Protection

UNIQUE SYSTEM ID NO
ATTRIBUTES
INFORMATION ASSETS USAGE POLICY

- Have you defined policies that provide clear direction regarding the usage of information? (Y / N / NA)
  - If N what is the time required to do so
  - What is the estimated cost to do so?

- Priority Level (1-5)
- Have you implemented policies that provide clear direction regarding the usage of information? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
    - Priority Level (1-5)
- Have you ensured that the policies in respect of the usage of information are communicated throughout the organization to users in a form that is relevant accessible and understandable to the intended reader? (Y/ N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)
- Have you documented policies that provide clear direction regarding the usage of assets associated with your electronic the communication facilities? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)
- Have you ensured that the policies regarding the usage of assets associated with your electronic communication facilities are communicated throughout the organization to users in a form that is relevant, accessible and understandable to the intended reader? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)
- Have all the incumbent employees of the organization accepted the above mentioned implemented policies? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)
- Have you ensured that the incumbent employees have access to the acceptable usage policies implemented? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)
- Have you ensured that contractors acknowledge adherence to your implemented policies? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)
- Have you ensured that third parties that access your environment acknowledge adherence to your implemented policies? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?

- Priority Level (1-5)
- Do all parties adhere to the stipulated policies that define the usage of e-mail? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)
- Do all parties adhere to the stipulated policies that define the usage of the Internet(Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)
- Are all the employees aware of the acceptable use of the organization's policies guidelines and limits? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)
- Are your organization's policies regularly reviewed and updated to ensure that they address the various changing trends regarding the usage of assets and the electronic communication facilities? (Y / N / NA)
    - If N what is the time required to do so
    - What is the estimated cost to do so?
- Priority Level (1-5)

6 Benefits of the South African Model

There are a number of benefits available when applying the above model and these range from cost effectives, build-up on current business continuity plans, to easy implementation.

The above model focuses on the actual CII i.e. it assumes that within a critical installation not all ICT systems are critical but that there is a subset of critical systems. This approach results in cost savings on implementation and it also allows system owners to implement comprehensive ICT Continuity Plans inclusive of all system interdependencies.

Another benefit is that it builds on current business continuity plans by giving a bird's-eye view of all critical ICT systems and their relationship with each other and other non-ICT infrastructures. This approach allows the effective deployment of limited resources i.e. skills, finance and ICT monitoring and response capabilities. If a country has only a small CSIRT capability it can only focus on the most critical ICT systems.

Our approach can be implemented regardless of the progress made on the development of the National Security Strategy and the related Policies, the establishment of a CII protection Agency and others. In short, this approach is capable of giving developing country similar levels of CII protection as the developed countries.

The modeling technique (object modeling) applied in this approach is best suited for computerizing the management of the whole national CII network. It allows the

application of CII analysis techniques like the Analytical Hierarchy Process and different variants of neural networks in order to discover hidden interdependencies and predicting possible future failures and impact of these on the current dependencies.

It goes without saying that in this approach there is a chance of missing some of the incidents that may be deemed less critical in the identification process.

7 Conclusion

This paper highlights a number of issues in the CII protection arena. The focal point was on the development of a simple but comprehensive identification methodology. We introduce the South African model, characterized by resource constraints such as funding and skills as an alternative to the developed world models.

We acknowledge that this model is not a solution for all countries but it is a reasonable bootstrap for a country that has a backlog of many years of ICT/Cyber security preparedness.

References

1. Dunn, M., Wigert, I.: The International CIIP Handbook 2004: An Inventory and Analysis of Protection Policies in Fourteen Countries, Centre for Security Studies (2004)
2. Suter, M.: A Generic National Framework For Critical Information Infrastructure Protection (CIIP) By Manuel Suter, Center for Security Studies, ETH Zurich (2007), http://www.itu.int/ITU-D/cyb/cybersecurity/docs/generic-national-framework-for-ciip.pdf
3. Dunn, M.: The socio-political dimensions of critical information infrastructure protection (CIIP). Int. J. Critical Infrastructures 1(2/3), 258–268 (2005)
4. Electronic Communications Security Pty (Ltd) Act (Act 68 of (2002), http://www.info.gov.za/view/DownloadFileAction?id=68106
5. Electronic Communications and Transactions Act (Act 25 of 2002), http://www.acts.co.za/ect_act/
6. National Key Points Act (Act 102 of 1980), http://www.midvaal.gov.za/LinkClick.aspx?link=NATIONAL+KEY+POINTS+ACT+102+OF+1980.doc&tabid=259&mid=893
7. Draft ITU National Cyber security/CIIP Self-Assessment Tool (2008), http://www.itu.int/ITU-D/cyb/cybersecurity/projects/readiness.html
8. Rinaldi, S.M., Peerenboom, J.P., Kelly, T.K.: Complex networks identifying, understanding and analyzing critical infrastructure interdependencies. IEEE Control Systems Magazine 21(6), 11–25 (2001)

Energy Theft in the Advanced Metering Infrastructure

Stephen McLaughlin, Dmitry Podkuiko, and Patrick McDaniel

Systems and Internet Infrastructure Security Laboratory (SIIS),
Pennsylvania State University, University Park, PA
{smclaugh,podkuiko,mcdaniel}@cse.psu.edu

Abstract. Global energy generation and delivery systems are transitioning to a new computerized "smart grid". One of the principle components of the smart grid is an advanced metering infrastructure (AMI). AMI replaces the analog meters with computerized systems that report usage over digital communication interfaces, e.g., phone lines. However, with this infrastructure comes new risk. In this paper, we consider adversary means of defrauding the electrical grid by manipulating AMI systems. We document the methods adversaries will use to attempt to manipulate energy usage data, and validate the viability of these attacks by performing penetration testing on commodity devices. Through these activities, we demonstrate that not only is theft still possible in AMI systems, but that current AMI devices introduce a myriad of new vectors for achieving it.

Keywords: AMI, Smart meter, Penetration testing, Attack tree.

1 Introduction

The smart grid being globally deployed today will forever change the way energy is used. This new infrastructure offers more efficient, lower cost, and more environmentally sound energy management than its antiquated predecessor. The advanced metering infrastructure (AMI) is a crucial piece of this new smart grid infrastructure. AMI provides a computer-based sensor system that extends from the homes and buildings that use power to the utilities that manage it. From a technology standpoint, AMI provides the necessary communication and control functions needed to implement critical energy management services such as fine grained pricing schemes, automatic meter reading, demand response, and power quality management. The smart grid has been widely deployed in Europe and Asia, with other parts of the world seeing more gradual but accelerating adoption.

The smart grid, AMI in particular, introduces new security challenges [1]. By necessity, AMI will consist of billions of low-cost commodity devices being placed in physically insecure locations. The equipment is under the control of the often disinterested, unsophisticated, or sometimes malicious users. Even in simple and/or low value services, such an arrangement would be extraordinarily difficult to secure.

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 176–187, 2010.

This paper considers one of the most important threats against the new smart grid—energy theft. Referred more generally as *theft of service*, energy theft occurs when a customer manipulates the energy usage statistics provided to the utility. To understand this threat, we develop an attack tree that systematically articulates the ways by which an adversary may attempt to manipulate usage data as it is collected, stored, or transmitted. We further preliminarily show the feasibility of different classes of attacks by penetrating currently deployed smart-meters, and attempt to identify root causes of existing vulnerabilities.

Theft of service for electric meters is nothing new. Annual losses in the United States alone are estimated at $6 billion [2]. Traditional theft in pre-AMI systems required the mechanical manipulation of analog meters. Conversely, in AMI, usage data may be tampered with after recording or in transmission to utilities. Moreover, software based attacks often require less expertise to execute and thus are likely to be more widespread. Precedence has shown and as we argue throughout, that these types of software attacks are quickly monetized by criminal groups that sell the hardware and software needed for theft of service. Examples include the descrambler boxes that lead to over $4 billion in cable theft per year [3] and sites that sell SIM unlock codes for cellular phones [4]. For these reasons, it is imperative for the AMI vendors, energy producers and distributors, governments, and customers to understand the potential scope and source of energy theft. This paper attempts to inform this need.

The rest of this paper is organized as follows. Section 2 explains the capabilities of AMI systems. Section 3 presents a threat model that shows how these capabilities may be leveraged for energy theft. A description of the tested equipment is given in section 4, and in section 5, we show the results of our security evaluation of a smart metering system, and explain their significance to energy theft. Finally, we discuss future directions for our study in section 6 and conclude in section 7.

2 AMI Background

The advanced metering infrastructure (AMI) is the sensor network of the smart grid. It provides the information about energy usage (demand) to utilities, consumers and the grid itself. This enables all parties to make better decisions about reducing costs and strain on the grid during times of peak demand. The necessary information about demand is coupled along with the energy distribution itself. This information is measured and aggregated by *smart meters*, digital electric meters that contain commodity CPUs, storage, and communication interfaces. These two components–smart meters and communication networks–form the infrastructure needed to provide AMI services. Broadly speaking, smart meters perform four basic functions with respect to power management; *a*) the monitoring and recording of demand, *b*) the logging of power relevant *events*, e.g., outages, *c*) the delivery of usage and logging information to the upstream utilities, and *d*) delivering and receiving of control messages, e.g., controlling smart appliances, remote disconnect, etc.

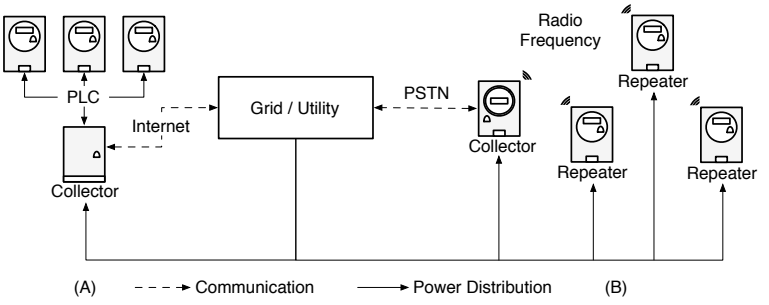


Fig. 1. Two example AMI network configurations. In (A) a power line communication (PLC) local network connects meters and a dedicated collector node. The collector communicates with the utility over the Internet. In (B) meters act as radio frequency repeaters to a collector node which itself functions as a meter. A backhaul link to the public switched telephone network connects the collector and utility.

AMI enables a number of services related to demand measurement and billing. Meters supporting automatic meter reading (AMR) can report demand to utilities automatically via communication networks. The two predominate network configurations are shown in Figure 1.[1] A local network of *repeater* meters is established between meters for the purpose of aggregating usage information to a *collector* meter. A backhaul network is then used to to transport the aggregate data from the collector to the utility. Local networks typically rely on either wireless mesh topology or power line communications (PLC). The backhaul link often uses a public network such as the Internet or the public switched telephone network (PSTN). Along with AMR for meter reading, AMI provides additional capabilities for dynamic pricing schemes such as time-of-use.

Time-of-use (TOU) [5] pricing refers to a pricing scheme in which power costs more during hours of peak demand. TOU schemes divide a day into several partitions called tariffs, typically peak and off-peak. Ideally, customers will be motivated to reduce costs by moving some energy-intensive tasks to off-peak hours, reducing the peak strain on the grid.

Beyond new interactions with customers and the grid, smart meters promise new anti-tamper measures. Previous meter tamper detection mechanisms were limited to locks and tamper-evident seals. While these measures are often sufficient for keeping honest people honest, they offer little to deter malicious tampering, and are often circumventable. Beyond physical tamper detection mechanisms, smart meters may be configured to log events concerning command history and conditions in the meters environment. This includes the detection of events indicative of physical tampering. One such mechanism, outage

[1] Other configurations are available but less widely deployed, such as those which use cellular networks. Apart from the access media, the operation is identical to that described in this paper. Thus the majority of attacks described herein apply to AMI systems built on these other access networks as well.

notification, records periods during which voltage has dropped or been removed from the meter's sensors. Reverse energy flow, which is indicative that the meter has been inverted in its socket, may also be detected through meter firmware. If customers participate in distributed generation, reverse energy flow is no longer indicative of meter inversion. In this case, additional tamper detections are necessary to differentiate between meter tampering and power actually received by the grid. Having covered the relevant AMI functionality, we go on to describe the threat model for energy theft.

3 Energy Theft

In this section, we use the security modeling technique of attack trees [6] to understand strategies for energy theft in AMI. Attack trees recursively break down an adversary goal into subgoals until a number of possible attack strategies are reached. The root node specifies the single goal of all attacks in the tree, in our case, this goal is demand forgery. Below the root node is a set of sub goal nodes that describe different approaches towards the root goal. The leaf nodes, which have no descendants, represent the specific attacks that must take place for the goal to be achieved. Paths to the root goal are augmented with the logical operators AND and OR which determine whether one or all of the children in a given internal node need be completed in order to achieve the goal.

The attack tree model is well suited here for a number of reasons. Individual attack trees can be composed to achieve specific goals. For example, an adversary that attempts to cause rolling blackouts [7,8] may have a sub-goal of forging energy demand at distribution substation meters. Attack trees also provide a way to reason about a system's vulnerable points in a top-down manner. This is useful for identifying both the root causes of attacks as well as the "low-hanging fruit" that is likely to be exploited.

3.1 Attacker Model

Before describing the attack tree for energy theft, we define the types of attackers that are motivated to commit energy theft.

- **Customers:** Traditionally, customers have been the primary adversaries aiming to steal power. The means and motivation to tamper with analog meters is very much individual in nature. That is, customers are limited in their resources and technical abilities, but in the case of AMI, can distribute some of the labor in discovering vulnerabilities and designing attacks to members of organized crime.
- **Organized Crime:** The motivation in the case of organized crime is the monetization of energy theft. Because of the extended computing and network capabilities of AMI, the task of creating software and hardware tools to compromise smart meters can be offloaded from customers to professional hackers. Members of this group will leverage certain design aspects of AMI

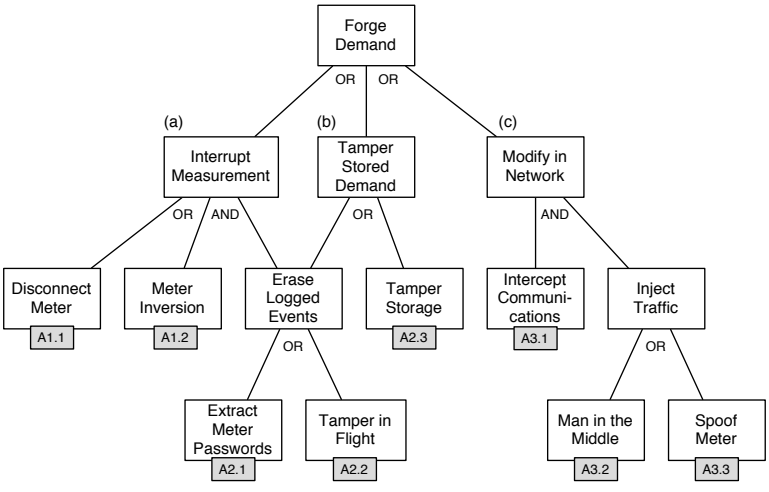


Fig. 2. An attack tree with leaves detailing the necessary attacks to commit energy theft. Theft can take place (*a*) before the meter makes a demand measurement, (*b*) before/while demand values are stored in the meter or (*c*) after measurements and logs have left the meter in transmission to the utility. The attacks leading to each of the three are labelled Ai.j for the jth attack that occurs at place (i).

systems, such as the widespread use of the same password set over many meters, to greatly amplify the profit from cracking a single meter.

- **Utility Insiders:** Utility insiders are implicitly trusted to be honest in the case of analog meters and the same model applies for AMI. It is preferable however that utility side systems enforce proper user and group management to provide properties such as separation of duties [9].
- **Nation States:** Adversaries with this level of expertise and resources have little motivation to commit theft as evaluated in this paper. They may however use discovered smart meter vulnerabilities for the denial of service or invasions of privacy.

3.2 Energy Theft Attack Tree

We present an attack tree for energy theft in Figure 2. As shown, the single requirement for energy theft is the manipulation of the demand data. There are three ways to tamper with demand data; *a*) while it is recorded (via electromechanical tampering), *b*) while it is at rest in the meter, and *c*) as it is in flight across the network. We discuss each of these ways in detail.

The first class of attacks, which aim to prevent the meter from accurately measuring demand, are the only class that previously existed for analog meters. The other two classes are exclusive to AMR and AMI. AMI does increase the difficulty of executing this class of attacks by logging sensor data that determines when power is cut to the meter, or if reverse energy flow occurs. Thus, to execute

attacks A1.1 or A1.2 undetected, it is necessary to also erase the logged events that indicate outage or reverse energy flow before they are retrieved by the utility. As these events are stored in the meter along with demand measurements, their removal falls under the second class of attacks on data stored in meters.

Smart meters store a large range of data. This includes tariffs for TOU pricing, logs of both physical events and executed commands, recorded net demand and their own programs among others. Because effectively all aspects of a smart meter's behavior are controlled by the contents of its storage, the ability to tamper with that storage (A2.3) gives a customer complete control over its operation. An attack involving overwriting the meter's firmware, while powerful, is a significant reverse engineering task. Thus, this type of attack is limited to members of organized crime aimed at selling meter hack kits.

For the purposes of energy theft, only a few select items in the meter's storage are of interest, namely, audit logs and the recorded total demand. Both of these values can be accessed through established administrative interfaces which require passwords. Their modification is usually limited to reset, clear in the case of an audit log and zero in the case of demand record. Consider the case in which a malicious customer has somehow obtained a meter password. The customer's electric bill may be reduced by X% by executing a demand reset operation after the first X% of the billing cycle. Because the administrative interface to the meter requires login credentials, a prerequisite to these attacks is extracting the necessary passwords from the meter (A2.1). In section 5 we explain one method that can be used for extracting the meter passwords and explain the far reaching consequences once they are no longer secret.

The third class of attacks involves injecting forged values into communication between meters and utilities. These attacks contribute to the above described monetization of energy theft in that they may be executed by any node between the meter and utility, which is not necessarily at the site where the meter is located. Furthermore, because of the two tier architecture of AMI, (local networks and backhaul links), executing a network based attack at a collector node makes possible the modification of all demand recorded for the set of repeaters. In some commercially available AMI systems, this can be in upwards of 1,000 nodes depending on the particular metering system.

The goal for subtree (3) requires two distinct types of actions, interposition of the attacker on the backhaul network (A3.1) and injection or modification of traffic between the meter and utility (A3.2,A3.3). Interposition is needed for any passive attack, including capturing the protocol between meters and utilities for reverse engineering. Network interposition can most easily be achieved close to one of the endpoints. For customers, tapping a line between the meter and the first backhaul link is the easiest. Utility insiders would have ready access to the links and routers leading up to the computers performing remote meter reads. The second task, traffic injection, requires the attacker to replace demand information from meters with forged data. In the event that an AMI system correctly uses cryptography for message integrity and authentication, this attack will require that the keys used for encryption first be extracted from meter storage (A2.3).

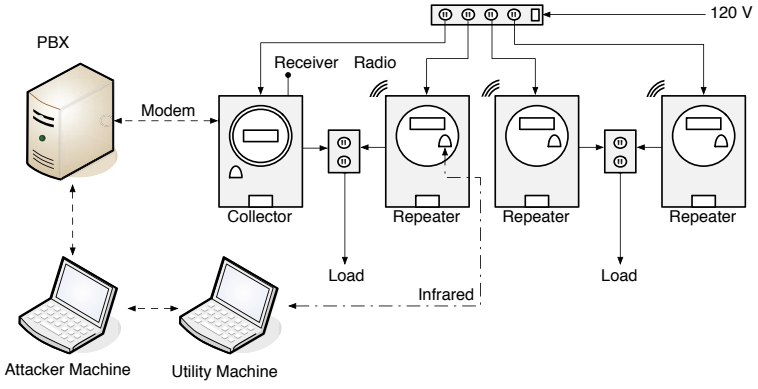


Fig. 3. Experimental testbed

If there is a flaw in the authentication or integrity protocols between the meter and utility, then a meter spoofing attack (A3.3) is sufficient for sending forged demand data and event logs. In this attack, a common device, such as a laptop computer, is used to receive calls from the utility in place of the meter and provides crafted values for specific fields. If the authentication mechanism is flawed but an encrypted channel is established between the meter and utility, a "man in the middle attack" (MIM) will be required [10]. This involves a node on the backhaul link from the meter to the utility to impersonate one to the other while the secure session is established to obtain the key used for cryptographic message integrity.

4 System under Study

In this section, we describe the environment and tools used for our preliminary smart meter security analysis. This analysis included reverse engineering and attacking meter communication protocols and details about the capabilities of the meters themselves. We describe the functionality and aspects of the implementation that are relevant to the results of the security analysis without respect to any specific vendor or equipment.[2]

The full experimental testbed is shown in Figure 3. It provides the full range of functionality needed to evaluate the security of the meters and communications within a typical AMI configuration. The local network, a wireless mesh operating in the 900 MHz band, is the only interface not yet evaluated in our study. The PSTN is used for the backhaul network to the utility. An Asterisk [11] based

[2] We are currently working with the appropriate agencies to notify the vendors of our preliminary findings. Due to the potential impact on public safety, it has been suggested that we do not identify the specific vendor with the current text. It is our expectation that we will be able to release this information in the final proceedings version with more detail on the tested equipment and attacks presented below.

private branch exchange (PBX) is used to simulate the telephone network for communication between the meter and utility. A "utility machine" runs the back-end software used by utilities for reading, programming and resetting meters. Both the utility machine and collector meter communicate over telephone using voice-band modems.

The testbed meters are form 2S single phase residential meters. Each meter is equipped with an infrared optical port that can be used for the same functions as the modem port, albeit without the same security measures. Both the modem and optical ports require passwords to read measurements and modify meter programs. The configuration has three repeater nodes in the local network and one collector connecting to the utility machine.

Auditing is used for outage notification and reverse energy flow detection as well as other more benign events. The audit logs may be retrieved via the optical port or telephone modem. Note that as some customers have contracts to sell generated power back to the grid, the reverse energy flow detection can be disabled. The intrusion detection feature in the collector node ensures that the meter is the only device off the hook when communicating with the utility. The meter automatically interrupts communication and hangs up when another phone is picked up.

Additional monitoring software was run on the utility machine to capture both the telephone modem and optical port protocols. The meter reading software running on the utility machine was disassembled in order to understand the use of cryptography in communications as well as the structures used to parse protocol messages. A separate "adversarial machine" is used for attacks on communication between the meters and utility machine.

5 AMI Security Analysis

The results of our security analysis show that the studied system contains design flaws that allow for energy theft using both pre- and post-AMI techniques. For each type of vulnerability, we describe the existing protections, how they may be circumvented and either describe a proof of concept attack as validation or explain how how validation could be achieved in future work.

Physical Tampering – A number of the identified attacks (A1.1, A1.2, A2.1, and A2.3) require some type of physical tampering of meters. Two types of tamper detections are provided by the studied meters, physical and firmware based. The physical tamper protections are the same as those for analog meters. Tamper evident seals are essentially the only means of detecting that the meter enclosure has been opened. Typically, two types of seals are used, one on the meter socket enclosure provided by the utility, and one on the meter's outer cover. The utility seal is a non-standard serialized tag and is outside the scope of this study. The seal placed on the meter's cover is a standard aluminum meter seal with a flag containing a stamped string between one and five characters. We were able to pick any stamp of our choice (including those used by vendors and utilities) from the tamper seal vendor without any special credentials. The ability to replace the seal on the meter cover eliminates any evidence that the cover was removed.

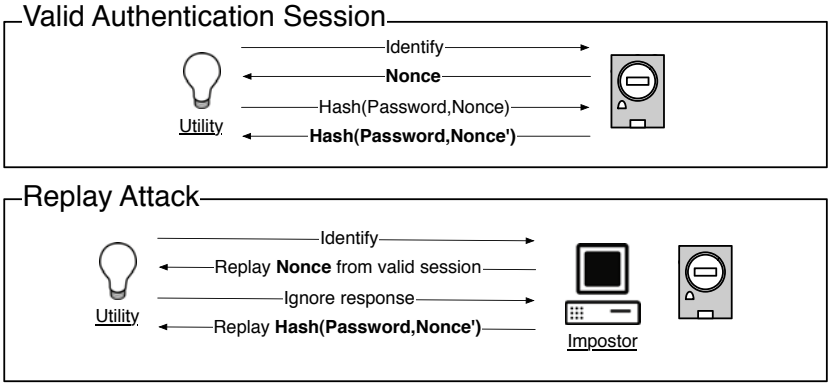


Fig. 4. The replay attack discovered in the studied system. Because the two messages in the mutual authentication round are dictated by the Nonce, replaying a previously recorded nonce will allow the impostor to authenticate without knowing the password used to key the hash.

Password Extraction – Attack (A2.3) requires that meter passwords be extracted. If the meter is physically tampered with, this may be achieved through optical port snooping. We used monitoring software on the utility machine to capture the optical port protocol used to communicate with the meter and found that passwords are transmitted in the clear. Once the meter is opened, placing a reader device on the optical port pins or near the optical lens is sufficient for capturing the password amidst the rest of the protocol. Furthermore, the password is always identifiable can be located according to the ANSI C12.18 standard [12].

Eavesdropping – The studied meters offer some protection against the interception of traffic (A3.1) over the telephone backhaul link. Whenever a meter detects that another device on the same line has gone off the hook, it will hang up. This works correctly in the case when a device such as a telephone or modem (technically any FXO [13]) picks up on the line. This feature does not work in the case of devices placed on the path between the meter and first link to the phone company. The PBX in our experimental testbed is one example of such a device (technically an FXS [13]). Using the open source software running on the PBX, we were able to monitor modem communication. In a full attack scenario, the monitored communication would later be demodulated from a waveform to the actual protocol.

Meter Spoofing – One risk of placing a physically insecure device on a network is the potential for a spoofing attack (A3.3), in which another adversarial device impersonates the legitimate one. The studied system uses the standard ANSI C12.21 protocol for mutual authentication of meters and utilities. In this protocol, the meter creates a cryptographic nonce which is sent to the utility. The

utility software then computes a message authentication code (MAC) by hashing the password and nonce. The calculation is done using the ANSI X3.92-1981 data encryption algorithm [14]. The MAC is then sent to the meter which calculates its own MAC which is sent to the utility software. At this point, mutual authentication is complete. The flaw in the studied system is that the utility software does not verify the freshness of the nonce from the meter. Thus, an adversary that is able to eavesdrop on an authentication session can replay the nonce and authenticate itself as the meter. See figure 4.

For a proof of concept meter spoofing attack, we used a laptop computer to impersonate a meter to the utility. Using the communication logs from eavesdropping on the backhaul link, we wrote a computer program to answer the utility's call and perform a demand and diagnostic read function. Using this program, we were able to insert chosen values for any field ready by the utility, including demand. Note that in the event that meters are modified to use encryption and mutual authentication, meter spoofing may still be achieved by extracting the cryptographic keys from the meter's storage (A2.3).

6 Understanding Vulnerabilities

Up to this point, we have modeled attacks leading to energy theft and shown vulnerabilities and proof of concept attacks in an AMI system. The goal now is to understand the design assumptions behind the vulnerabilities. The grouping of attacks by these assumptions is shown in Table 1. We explain the impact of each of these assumptions on attacks on AMI and show that they create three properties that increase the ease and monetization of energy theft. These are, amplification of efforts, division of labor, and an extended attack surface.

Table 1. A summary of the vulnerabilities in the studied AMI system and the attacks they enable. The specific vulnerability in the studied system that enables each attack identified in section 3 is shown along with the design assumption behind that vulnerability. Assumptions are numbered for reference in this section.

Attack Number & Description	Vulnerability	Design Assumption
(A1.1, A1.2) Measurement interruption	Inadequate physical tamper protections	1. Physical limitations
(A2.1) Password extraction	Insecure optical communication	2. Near field security
(A2.3) Meter storage tampering	No Firmware Integrity Protections	3. Physical integrity of meter
(A3.1) Communication interception	Insufficient intrusion detection	4. Trusted backhaul nodes
(A2.2, A3.2, A3.3) Communication tampering	Failure to check for replay	5. Trusted endpoint node

As is the case with analog metering, assumption 1. states that there are economical and practical limitations to how well a meter can be physically secured. This limitation ideally would be addressed by the advanced security features provided by AMI. This however is not the case as the existing firmware protections are not tied to the meter's physical enclosure. While possible electromechanical tampering is detected, the assumption is made that the meter internals, and thus the tamper detection mechanisms, are not accessed.

Assumption 2., which states that optical port communication with an untrusted device is secure, is incorrect. Given this, both the optical signal could have been sensed by special equipment or recorded inside a compromised meter. While special equipment may be needed to obtain a password from the optical port, the payoff is multiplied by the number of meters using that password.

Assumption 3. is another example of amplification of adversary effort. The potential for tampering with the meter's stored firmware has several consequences beyond the mere ability to steal power. First, modifications at the firmware level are hard to detect without off-line inspection of the firmware contents. Second, the customer using the tampered firmware for theft does only the small amount of work needed to upload the malicious image. The majority of effort is put forth once by a group with the goal of selling malicious meter firmware. This is indicative of both the ability of AMI to multiply the effort put into attacks and a distribution of labor between organized crime and customers. Finally, as tampered firmware may be a sign of remote exploitation, the customer has plausible deniability if tampering is detected.

Assumption 4. leads to the poor use of authentication and encryption for integrity as well as a circumventable intrusion detection mechanism in the studied system. This assumption is likely due to misunderstanding the security requirements of extending the attack surface into public networks.

Finally, the failure of mutual authentication of utilities and meters, by assumption 5., creates a vulnerability that is both widespread and easily exploitable. The ability to simply substitute another device for a meter encourages the creation and distribution of meter spoofing software which leaves no evidence of tampering at the meter itself. This is another example of AMI's extended attack surface.

7 Conclusion

We posit that the basic requirements of AMI are in conflict with security. While some poor engineering choices are sure to exacerbate some of these issues, there are fundamental reasons why a fully digitized metering system is inherently more dangerous that its analog predecessor. Several of these reasons include:

1. **Amplification of effort:** In many cases, compromising a single meter is sufficient for stealing power with many more. Attacks that capture a password once and use it many times or the penetration of a head end meter to modify all usage in an area are exemplary.
2. **Division of labor:** Customers may avoid a large degree of risk and effort by using pre-made meter programs to overwrite meter firmware and spoof

communications with utilities and the grid. It is a near certainty that 'script-kiddie' style attacks on meters will be easily attainable off the Internet.

3. **Extended attack surface:** AMI extends the attack surface for metering to entire public networks. Tampering at the endpoints of these networks is particularly useful for energy theft as the demand information for many meters passes through collector meters and links to utility servers.

Note that this list is in no way comprehensive.

In this paper, we studied the ways in which energy theft is likely to occur in AMI systems. Our findings show that those interested in mounting theft will have the capacity to do so, at least in the foreseeable future. What is left is to begin working now in further identifying the vulnerabilities and finding ways to mitigate them. Such efforts should not only be targeted to combating theft, but also to all of the other personal and national safety issues vulnerabilities in AMI systems represent. It is our hope that this paper has provided a usable roadmap to begin these efforts.

References

1. McDaniel, P., McLaughlin, S.: Security and Privacy Challenges in the Smart Grid. IEEE Security & Privacy Magazine (May/June 2009)
2. Electric Light and Power Magazine: Reducing revenue leakage (2009), http://uaelp.pennnet.com/
3. National Cable Television Association: Ncta 2005 signal theft survey (2005), http://www.ncta.com
4. Netwondo LLC: Unlock your google phone (2009), http://www.unlock-tmobileg1.com/
5. King, C.S.: The economics of real-time and time-of-use pricing for residential consumers. Technical report, American Energy Institute (2001)
6. Schneier, B.: Attack trees. Dr Dobb's Journal 24(12) (December 1999)
7. Kinney, R., Crucitti, P., Albert, R., Latora, V.: Modeling cascading failures in the north american power grid. The European Physical Journal B - Condensed Matter and Complex Systems 46(1), 101–107 (2005)
8. Liu, Y., Ning, P., Reiter, M.K.: False data injection attacks against state estimation in electric power grids. In: Proceedings of the 16th ACM Conference on Computer and Communications Security (November 2009)
9. Clark, D.D., Wilson, D.R.: A comparison of commercial and military computer security policies. In: IEEE Symposium on Security and Privacy, pp. 184–195 (1987)
10. Desmedt, Y.: Man in the middle attack. In: van Tilborg, H.C.A. (ed.) Encylopedia of Cryptography and Security, p. 368. Springer, Heidelberg (2005)
11. The Asterisk Project: Asterisk open source pbx, http://www.asterisk.org
12. American National Standards Institute: C12.18 Protocol Specification for ANSI Type 2 Optical Port (2006)
13. 3CX: FXS, FXO Explained (2009), http://www.3cx.com/PBX/FXS-FXO.html
14. American National Standards Institute: ANSIX3.92-198 Data Encryption Algorithm (1981)

Current Capabilities, Requirements and a Proposed Strategy for Interdependency Analysis in the UK

Robin Bloomfield[1,2], Nick Chozos[2], and Kizito Salako[1]

[1] Centre for Software Reliability, City University London. 10, Northampton Square, College Building, EC1V 0HB, London, UK
{reb,ptp,kizito,dw}@csr.city.ac.uk
[2] Adelard LLP. 10, Northampton Square, College Building, EC1V 0HB, London, UK
nc@adelard.com

Abstract. The UK government recently commissioned a research study to identify the state-of-the-art in Critical Infrastructure modelling and analysis, and the government/industry requirements for such tools and services. This study (Cetifs) concluded with a strategy aiming to bridge the gaps between the capabilities and requirements, which would establish interdependency analysis as a commercially viable service in the near future. This paper presents the findings of this study that was carried out by CSR, City University London, Adelard LLP, a safety/security consultancy and Cranfield University, defense academy of the UK.

Keywords: Critical Infrastructures, Interdependency modelling and analysis, R&D strategy.

1 Introduction

The UK Centre for the Protection of National Infrastructure (CPNI), the Technology Strategy Board (TSB) and the Engineering and Physical Sciences Research Council (EPSRC) commissioned a feasibility study to identify the state-of-the-art in Critical Infrastructure (CI) interdependency modelling and analysis and to develop a strategy for research and practice, aiming to bridge the gaps between existing capabilities and Government/industry requirements.

The study, carried out by the Centre for Software Reliability of City University, London, Cranfield University, Defense Academy of the United Kingdom and Adelard LLP resulted in two publically available reports:

- The 'main' report [1], which presents the overview of capabilities, requirements and the proposed strategy.
- A secondary report [2], which is an introductory research review in the areas of modeling, analysis and visualization of infrastructure interdependencies.

This paper will briefly present the study, discuss some of its findings, and conclude with the proposed strategy.

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 188–200, 2010.

2 Background: The Cetifs Study

The Cetifs (CPNI, EPSRC, TSB Interdependency analysis Feasibility Study) methodology comprised the following activities:

1. Analysis of two recent major UK multi-infrastructure disasters: The Buncefield explosion [4] and the 2007 floods [5].
2. Consultations with a wide a range of Critical National Infrastructure (CNI) stakeholders (government, industry and academia)
3. A review of research specific to modeling and analysis of dependencies in CIs (in a separate report, [2]).
4. A questionnaire survey based on the three previous activities distributed to utility companies IT and security departments

2.1 The Buncefield Explosion

The explosion that took place at the oil storage depot located in Buncefield in December 2005 has been characterized as the biggest explosion in peacetime Europe. The explosion affected the operation of multiple infrastructures (energy distribution, transportation, information infrastructure, finance, health as well as the environment). This incident is of particular importance as it unveiled some important issues with regard to information infrastructures (II).

We mainly focused our analysis on an IT company/data centre named Northgate Information Solutions, which was severely affected by the explosion. The servers that were at these premises hosted patient records and admission/discharge for a number of hospitals in the area, a North London payroll scheme of approximately £1.4 billion, and systems/data for several local authorities [4] among others.

2.2 The 2007 Floods

The floods that struck much of the country during June and July 2007 were extreme, affecting hundreds of thousands of people in England and Wales. It was the most serious inland flood since 1947 [5]. 13 people lost their lives, approximately 48,000 households and nearly 7,300 businesses were flooded and billions of pounds of damage were claimed. In Yorkshire and Humberside, the Fire and Rescue Service launched the "biggest rescue effort in peacetime Britain".

The floods affected multiple infrastructures, such as water and food supply, power, telecommunications and transportation, as well as agriculture and tourism. Many businesses also suffered flooded sales premises, together with damage to stock and equipment.

2.3 Incident Analysis Conclusions

The analysis of these incidents helped us to understand some of the challenges that infrastructure owners and the government are facing. We found that there are several issues which, although they are known, they are not well understood. These served as a basis for our consultations and were the following:

Geographical dependencies are, to a certain extent, known, as the identification of physical proximity of assets is straightforward, especially when we consider an area surrounding a plant or within a flood-vulnerable area. Nonetheless, there were several surprises in these events (*e.g.*, during the floods, several critical services had to be shut down for precaution in case the flood reached them but there was uncertainty as to whether that was actually needed or not). There are also more complex and indirect consequences (*e.g.*, the effect the Buncefield explosion had on the adjacent business park and the data centre in particular was also deemed as a surprise).

Competition for resources. This challenge arises during an incident and can also lead to interdependencies or further cascade effects. Capacity and bandwidth of resources are known to infrastructure owners; however, during crises they may be reached very quickly, and in unusual ways. Competition for resources can also manifest when an asset that provides a resource is lost (*e.g.* a power station), where other dependent nodes will have to find alternative suppliers.

Long term effects. In some cases, major incidents can involve significant long term loses to infrastructure and economy by complex cascade paths. One typical aspect of this is the effect a disaster can have on tourism. In the Pitt review there was an extended discussion on the role of media following the floods and the long-term effect on tourism and the economy of affected areas. Although there are a number of studies in macro-economic impact of infrastructure failures, the long term effects of such disasters and how they can be controlled are aspects that are not well understood and require more detailed analysis, considering various parameters such as the role of media.

We also concluded that there is a *lack of empirical data* to support in-depth analyses that will help us understand interdependencies better. This is due to the comparative rarity of events, and the difficulty in attaining data from multiple organizations, with many incidents going unreported or kept as anecdotes within one infrastructure. As part of this study and continuing work with TNO [7] we are analyzing the implications of their large infrastructure incident database [1].

2.4 Consultations and Questionnaire Survey

The consultations formed the biggest part of this study; in particular, we carried out semi-structured interviews with:

- Parts of the UK government that are concerned with the prevention of and response to major CNI disruptions, resulting either from attack or natural disaster. These consultations helped us formulate the context of the study and the requirements that Interdependency Analysis (IA) services would have to satisfy.
- Private companies and research institutions that develop tools or use them to offer services that can assist in the identification of interdependency vulnerabilities. These stakeholders provided us the understanding of what the state-of-the-art is, and what capabilities can be offered currently.

Before discussing the requirements and capabilities, we ought to present the different perspectives that stakeholders have as these perspectives pose different sets of requirements and interests in IA. These perspectives have been organized around the

concept of resilience, as it provides a useful framework within which to consider different stakeholder approaches, requirements and responsibilities for CI services.

2.5 Perspectives on CNI Resilience

Interdependencies are often discussed as a source of threat to systems. Indeed this can be the case and in particular unforeseen interdependencies can be a source of surprise and uncertainty in our ability to understand risks and system behavior. However interdependency is also central to providing tolerance to attack and failure, a means for adaptation and overall resilience.

The loss of system capacity due to an incident can be seen as an indication of how resilient a system can be. This viewpoint is shared by the US Department for Homeland Security (DHS) and UK Resilience. This resilience perspective is shown in Figure 1 below.

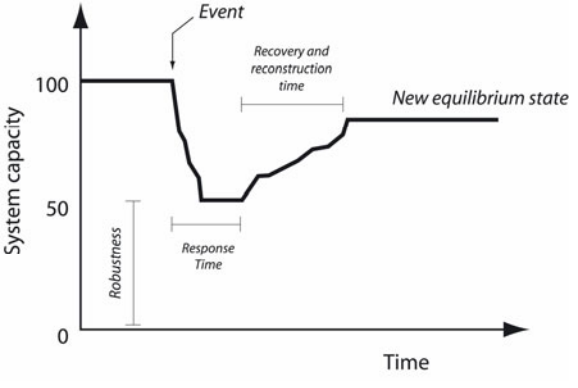


Fig. 1. Resilience

However in [6] the emphasis is on the ability of a system to adapt and respond to changes in the environment. In a recent report for the Defense Science and Technology Laboratory (DSTL) [3] produced by CSR, City University London, two types of resilience were distinguished:

- *Type 1: Resilience to design basis threats*. This could be expressed in the usual terms of availability, robustness, etc.;
- *Type 2: Resilience to beyond design basis threats*. This might be split into those known threats that are considered incredible or ignored for some reason and other threats that are unknowns.

Some policies consider an "all hazards" approach that addresses both malicious and accidental attacks on systems. In addition, the notion of *dependability*, or dependability and security, as an umbrella term is useful to capture the need to address all attributes (safety, security, availability etc.) rather than just a single attribute.

The overall service level view is summarized in Table 1 below:

Table 1. Phases of resilience

Phase	Action to increase resilience
Preparation and learning	Reduce frequency of events by early warning and upstream measures. Provide early warning, operator support. Learning from experience (major incidents, minor mishaps, near misses), training.
Initial loss	Increased robustness by - Network design addressing topology, redundancy, diversity. - Classification of critical nodes and suitable hardening. - Understanding of events and scenarios
Detection	Communication between services. Variety of forecasting approaches. Detection of compromises.
Decision	Situational awareness. Planning and training (scenarios) and use of synthetic environments.
Recovery	Resource deployment; dependent assets identified - Awareness state of other networks. - Communication and co-ordination.

The different stakeholders all had an interest in resilience but had very different emphases. Broadly speaking these concerned the scope of their responsibilities, whether it was:

- *All hazards approach*: all hazards are considered, including both natural disasters and malicious attacks;
- *Security and vulnerability focus*: identification of security critical assets and consideration of vulnerabilities/threats to them;
- *Natural hazard focus:* only considers events such as floods/earthquakes and their effect on CNI;

And also the overall purpose of their analyses e.g.

- *Identification* of vulnerabilities (dependencies) in stable system state;
- *Incident response*, i.e., control of the incident and evacuation and coordination of emergency services;
- *Long-term effects and recovery* e.g., environmental, financial.

We can use the resilience-dependability framework to capture the different perspectives of stakeholders. For example, those of CPNI and the UK Home Office Civil Contingency Secretariat (CCS) are shown in the table below.

Table 2. CPNI and CCS perspectives

Framework component	Stakeholder: CPNI	Stakeholder: CCS
What services are addressed?	All within scope of NI suitably prioritized.	All
Which dependability attributes are concerned?	Classic security attributes – confidentiality, integrity, availability.	Emphasis on availability.
What range of hazards/ threats?	Security related only.	Natural hazards in terms of initiation. Advice from CPNI on security. All hazards in decision and recovery phases.
Which resilience phase?	Emphasis on prevention and preparation and learning phase. Advice to CCS during incidents.	National risk assessment deals with long term losses. Emphasis on recovery and incident response.
What services are addressed?	All within scope of NI suitably prioritized.	All

A security evaluation could then be seen as evaluation of resilience for certain threats (e.g. malicious ones) and for certain attributes (confidentiality, integrity, availability). The evaluation of the security part of resilience would then address the different stages of Table 2.

In this study we were particularly interested in (inter-) dependencies, and so we can use the framework to assess what dependability attributes, what resilience phase and what threat scope is of concern and being addressed by particular modelling and analysis approaches.

2.6 Questionnaire Survey

We further explored these issues with a small questionnaire survey that was targeted at utility companies IT and security managers.

From the responses we have received, we found that utility companies address the challenges of infrastructure interdependencies by ensuring close relationships with suppliers and vendors. They believe that close relationships can assist in understanding the various risks associated with their providers' failure and their overall level of resilience. Risks are monitored through internal risk review groups, and company boards oversee the results. Also in some cases utilities hold industry forums to exchange information, or engage in regular review meetings. Exercises involving suppliers have also been carried out. In some cases, alternative providers have already been sourced as part of contingency planning.

However, the protective measures to be taken depend on the nature of the risk or vulnerability and on the particular department. Overall, utility companies focus on

improving resilience by having business continuity planning, frequent risk assessment, back up systems (especially for IT), as well as security technologies.

Although infrastructure dependencies are considered in risk assessment, this is mostly done in more traditional ways, without tool support. In one case, it was suggested that mapping software was used, although just once, for examining proximity of functions to cable routes. In addition, none of the respondents were aware of any technical documentation, research or conferences in infrastructure interdependency, something which perhaps suggests the presence of a gap between research and practice.

Most responders suggested they had experienced either minor or major disruptions due to failure of other infrastructure providers.

The questionnaire also probed whether there was scope for some form of IA as a distinct service. There was no clear consensus from respondents; some believed it could be, and some suggested they would be interested if it was part of a wider, risk assessment service. The issues of trust and confidentiality were raised as serious obstacles.

2.7 Research Review

The models and simulations developed to support infrastructure modelling and simulation are diverse and complementary. There are multiple ways in which these models are related and there is no single taxonomy or classification that suits all purposes.

In the review we focus on the results of the models to provide a basis for describing relationships between them. The classification of modelling activities from this perspective, applied in particular to models, tools and methodologies is provided in [2]. This includes:

- *Abstraction level and model boundaries:* Questions such as "how much of the real world should be modelled?" constrain modelling methodology and the applicability of modelling results. A continuum of possibilities exists ranging from high-fidelity (very detailed) simulations to mid-range and low-fidelity models;
- *Technique and underlying theory:* (Inter)dependency analysis of complex systems has been recognized as an inherently interdisciplinary activity. There exists a wealth of experience and knowledge relevant for (inter)dependency modelling. This column in the table below gives information about established formalisms, theory and techniques used in building and analyzing the models;
- *Model applicability:* The type of problems where the model can provide useful support is indicated in this column and the extent of tool support.

The incident analysis, the consultations and the questionnaire survey helped us to formulate the requirements, while the research review and again the consultations helped us to evaluate the state-of-the-art, the current capabilities. Capabilities and requirements are discussed in the following two sections.

3 Initial Requirements

From our discussions with stakeholders we concluded that:

1. There is recognition that interdependencies are part of wider issues of understanding infrastructure interaction.
2. They are concerned that they lack knowledge of infrastructure interactions.

3. There is sufficient expert judgment, anecdotes and incident analysis to suggest that this lack of knowledge may present a significant risk or a missed opportunity for improving resilience at all stage of the resilience lifecycle.
4. They see many potential advantages in a more sophisticated approach to infrastructure modelling but at present they do not know under what circumstance these uncertainties are significant and so can not justify the required investments.

In discussion with stakeholders we identified requirements across various areas that relate to infrastructure interdependencies. These areas are the following:

- *Inherent infrastructure resilience—scope and overall methodology:* Perspectives here address the level of resilience that is built in to infrastructures and normal operation.
- *Infrastructure analysis and support:* The consultation identified a number of different possible service delivery perspectives.
- *Hazard and vulnerability identification and management:* Perspectives vary on the scope of hazards to be addressed or the approach to the management of systems.
- *Resilience phases:* Potential capabilities and requirements that concern the various phases of resilience.
- *Critical information infrastructures:* A greater focus is given in this study to CII.
- *Dependability of the modelling:* An integral part the development of tools and analytical services is to ensure that they are dependable. There will be a need to trust the results of infrastructure modelling and analysis and possibly integrate information from a variety of trusted and less trusted sources. There will therefore be a variety of confidentiality requirements on the modelling tools and supporting IT infrastructure depending on their application and mode of service delivery. Unless these confidentiality requirements are met the modelling activity could provide a threat.
- *Evidence of costs and potential benefits:* Cost and benefit issues have to do with costs of failure and benefits of IA.

4 Current Capabilities

Providers of infrastructure modelling and/or (inter-) dependency analysis are either government-endorsed organizations, or leading private technology solutions providers. Overall, they offer a diverse range of services. Our consultations have aimed at understanding their capabilities and market deployment approaches. These will then be related with, and contrasted to, the initial requirements in section 3.

Figure 2 presents the components that we have considered in this study (see [2] for more detail).

These components are explained as follows:

Data, information and knowledge. This refers to the data that is fed into the simulation. Data can be either static or live. For instance, simulators are often linked to live weather feeds, GPS and other forms of live data sources. Data acquisition and verification are important challenges as insufficient, incorrect or inaccurate data can result in a misleading analysis.

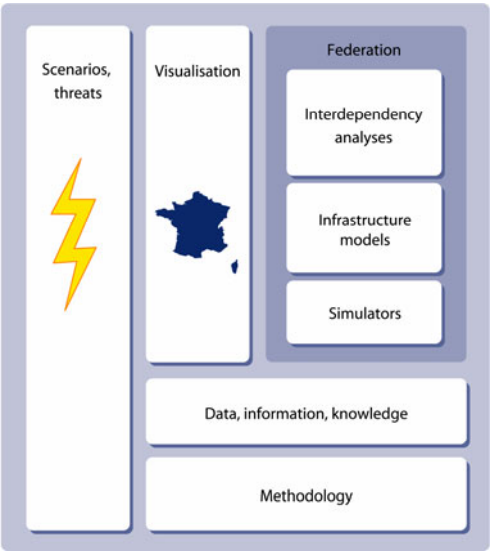


Fig. 2. Modelling components

Federation refers to the integration of several simulations (federates). This is primarily done through achieving interoperability among separately developed simulators. Standardization is required in order to define common elements.

Infrastructure models. Modelling within a single infrastructure or system is a diverse and mature field. Models are fundamental to understanding system behavior, evaluating risks and designing operational strategies.

Interdependency modelling can be considered according to the different perceived layers (e.g. of physical, control and supervisory management) and also in terms of a range of abstractions from high-level services to detailed implementations. For each of these abstractions, there are a wide range of possible modelling approaches and theories that can be deployed, ranging from qualitative models, stochastic activity networks to complexity science style models and high-fidelity simulations. These can be deployed at a varying levels of detail, e.g. to model the detailed implementation topology or to model the service topology and cascading effects.

Methodology. A defined and structured approach can assist in an efficient and effective modelling and analysis. The methodology contains aspects of requirement elicitation, data gathering and analysis, modelling, simulation and the eventual development of conclusions and decision support.

Scenarios and threats. Scenario development considers situations and sequences of events that are of particular concern, in order to identify threats and gain insight of the 'system' behavior under hazardous conditions. In most cases, a 'reasonably' worst case scenario is needed in order to focus planning and mitigation against a threat that has a realistic likelihood of occurring.

Simulators. Simulation is the imitation of some real thing, state of affairs, or process. There are many different types of computer simulation—the common feature they all

share is the attempt to generate a sample of representative scenarios for a model in which a complete enumeration of all possible states would be prohibitive or impossible.

Visualization refers to the graphical representation of the modelling and analysis. This can be either on a standalone PC screen, or on large, operating room screens, or over a set of various screen types, sometimes even distributed across various locations. Geographical Information Systems (GIS) are a typical example of visualization. In IA, visualization tends to be layered, with several filtering options to guide decision support and communication.

5 The Importance of "intangible" Infrastructures

One significant result from our consultations is the importance of "soft" intangible critical infrastructures, e.g. *trust and confidence* within society both in their own right and as an important component that is essential to the functioning of critical services. For instance, trust between individuals, between individuals and organizations and between these and the representative of the state is essential for the delivery of service. This, as with so many of the infrastructures, is often hidden but comes to the fore in times of crisis and recovery from disaster.

Trust is an asset that can be built-up, destroyed, squandered and undermined as with so many other assets and resources. If we are to assess interdependencies we need to take into account these essential yet softer aspects and their relationship to the more tangible aspects. Such assessment should appreciate that these soft aspects are just as much the target of security threats as the more obvious physical and cyber systems. Indeed, it may be that a patient and well read adversary would have a strategy that targets these assets. For example, the financial infrastructure relies very heavily on trust in the banking system for it to function at all. Witness the latest credit crunch, the Northern Rock bank crisis and also public trust in government announcements and the panic buying of petrol because people did not believe assurances about supply. An adversary strategy that relies on people legitimately taking their money out of a bank is far more effective than any physical raid on the bank (unless one wants to get rich). At a micro-level, social engineering attacks that exploit people's willingness to give passwords away can be seen as a form of attack exploiting confidence.

While in the past the soft infrastructure might have been separable from the more technical infrastructures they are clearly related. Trust in the competence of government and authorities is dependent on how well they cope with crises and incidents in both the physical and soft infrastructures. Moreover trust relationships that citizens have between themselves, organizations, government and agencies are strongly dependent on the information infrastructure: a trend that is likely to increase (see the UK transformational government agenda [8]).

Assets such as trust and privacy within society are important and can be seen as emergent properties; although they are affected by local aspects of trust they have a complex relationship to localized issues. Trust in organizations and government may exhibit the classic complex systems phenomenon of rapid transitions and "tipping points".

Understanding the role of trust and confidence in the protection of CI to the extent where it can be taken into account in CI modelling is arguably a great challenge. This is an active research area (e.g. [10]) but, most work is focused on the application of

trust models for the development of trusted IT networks, e.g. for information sharing, but the wider implications of trust seem to be under investigated currently.

6 A Proposed Strategy

The final part of the study was a gap analysis between the requirements and capabilities identified (as discussed in sections 3 and 4 respectively) to identify whether further research and development might be required, and if so, what form it should take.

IA needs a sufficiently rich model for the analysis to discover and assess the risks:

- Societal aspects need assessment as they provide possible hidden sources of commonality;
- Modes of operation have to be rich enough. These should include degraded modes of operation as they can amplify risks as levels of redundancy assumed at design time become defeated;
- Non-linearities in failure models (e.g. increased failure rates due to stress from nodes in the same locality) can lead to escalation and cascading effects.

We have identified four main potential capabilities:

- To provide specialized security analysts with a means for the assessment of interactions and interdependencies;
- To provide off-line support for risk assessors both aggregators of risk (as at CCS) and also individual infrastructure owners to evaluate the impact of dependencies and interdependencies;
- To provide off-line support for risk assessors both aggregators of risk and also individual infrastructure owners to evaluate the impact of dependencies and interdependencies during incidents (soft real-time);
- To provide real-time, decision support integrated command and control systems (hard real-time) that takes fully into account the impact of dependencies and interdependencies.

To address the required capabilities and gaps that we have identified the study proposed the following:

Trial state-of-the-art and emerging research. Develop and trial modelling approaches and decision-support tools and methodologies at various levels of detail. The trials would consider both qualitative approaches and off-line, soft real time and hard real-time infrastructure interactions. The modelling would consider functional, topological and probabilistic approaches. The trial should be sufficiently complex to enable scalability issues to be addressed and consider a number of different infrastructure mixes e.g.:

- Energy distribution (*e.g.* gas, electricity);
- Information infrastructures;
- Soft intangible infrastructures (*e.g.* trust, confidence).

The output of the exercise would be experience with the modelling approaches, assessment of costs/benefits and way forward and provide more clarity in current and future stakeholder requirements.

Real-time environment provide particular challenges and these should be addressed separately. Consider proposed future of decision support systems for key stakeholders and develop more detailed requirements to integrate interdependency approach.

Develop an interoperability approach to infrastructure modelling and analysis (e.g. by use of standards, interoperabilities, published Application Programming Interfaces (APIs)). This should promote both innovation and also a more componentized approach. Interoperability should cover behavioral models, topologies and associated data. Data costs can be significant and interoperability can provide an approach to amortizing data costs across applications.

Provide policy support and evidence base. Provide justification and focus of the programme, emphasizing the benefits and responsibilities for all stakeholders.

Define credible business models taking into account the fact that infrastructure and interdependency modelling has particularly close coupling to policy and to sensitive areas of risk assessment.

Offer knowledge transfer and coordination. Promote the research base and offer connection to practice by enabling interaction (e.g. via knowledge transfer activities), addressing costs of research and methodologies and developing a challenging research agenda.

Within each of these threads both natural hazards and security vulnerabilities need to be considered (e.g. by the emphasis in different scenarios).

7 Conclusion

This paper presented an overview of a study that was carried out by the Centre for Software Reliability of City University, Cranfield University, Defence Academy of the United Kingdom and Adelard LLP.

The study was based on consultations with a wide a range of Critical National Infrastructure (CNI) stakeholders (government, industry and academia) and a review of research specific to modelling, analyzing and overall understanding dependencies in infrastructures [1],[2]. The consultations and the research review identified to potential capabilities that would address current requirements and proposed a strategy aiming at achieving the capabilities that were identified as currently feasible.

References

1. Bloomfield, R., Chozos, N., Nobles, P.: Infrastructure interdependency analysis: Requirements, capabilities and strategy. Adelard document reference: d418/12101/3 issue 1 (2009), http://www.csr.city.ac.uk/projects/cetifs.html
2. Bloomfield, R., Salako, K., Wright, D., Chozos, N., Nobles, P.: Infrastructure interdependency analysis: an introductory research review., Adelard document reference D/422/12101/4 issue 1 (2009), http://www.csr.city.ac.uk/projects/cetifs.html
3. Bloomfield, R., Gashi, I.: Evaluating the resilience and security of boundaryless, evolving socio-technical Systems of Systems, research report fro DSTL, Centre for Software Reliability, http://www.csr.city.ac.uk/people/ilir.gashi/Papers/2008/DSTL/

4. Buncefield explosion official investigation website, http://www.buncefieldinvestigation.gov.uk/index.htm
5. The Pitt Review: lessons learned from the 2007, floods, official website (2007), http://www.cabinetoffice.gov.uk/thepittreview.aspx
6. Hollnagel, E., Woods, D., Leveson, N. (eds.): Resilience engineering: concepts and precepts. Ashgate Publishing Company (2006)
7. Toegepast Natuurwetenschappelijk Onderzoek (TNO), http://www.tno.nl
8. UK Cabinet Office, Cm 6683, Transformational Government, Enabled by Technology (November 2005)
9. Gosh, A., Del Rosso, M.: The role of private industry and government in critical infrastructure protection (1999), http://gost.isi.edu/cctws/delroso-ghosh.PDF
10. International Telecommunication Union, Creating Trust in Critical Infrastructures workshop (2002) http://www.itu.int/osg/spu/ni/security/

Stochastic Modelling of the Effects of Interdependencies between Critical Infrastructure

Robin Bloomfield[1,2], Lubos Buzna[3], Peter Popov[1], Kizito Salako[1], and David Wright[1]

[1] Centre for Software Reliability, City University London. 10, Northampton Square, College Building, EC1V 0HB, London, UK
{reb,ptp,kizito,dw}@csr.city.ac.uk
[2] Adelard LLP. 10, Northampton Square, College Building, EC1V 0HB, London, UK
reb@adelard.com
[3] University of Zilina, Univerzitna 8215/1, 010 26 Zilina, Slovakia
lbuzna@ethz.ch

Abstract. An approach to Quantitative Interdependency Analysis, in the context of Large Complex Critical Infrastructures, is presented in this paper. A Discrete state–space, Continuous–time, Stochastic Process models the operation of critical infrastructure, taking interdependencies into account. Of primary interest are the implications of both model detail (that is, level of model abstraction) and model parameterisation for the study of dependencies. Both of these factors are observed to affect the distribution of cascade–sizes within and across infrastructure.

Keywords: Interdependency Analysis, Critical Infrastructure, Cascade–size Distribution, Continuous – time Stochastic Process.

1 Introduction

Dependencies within and between *Critical Infrastructures* (CI) have been recognised as important for achieving (or undermining) acceptable system safety, security and dependability [1, 2]. There is a growing body of research into the quantitative modelling of Complex systems, their dependencies and the implications, thereof, for the occurrence and sizes of cascades [1, 3-6]. The need to understand (inter)dependencies is evidenced by the occurrence of spectacular, catastrophic *cascades*[1] as a direct result of dependencies. One such example is the North American Blackout that occurred on the 16th of August, 2003 which affected an estimated 10 million people [7]. Yet another example is the explosion that occurred on 11 December, 2005 at Buncefield Oil Storage Depot, Hertfordshire, in the United Kingdom. The explosion affected part of the local *Information Infrastructure* and, ultimately, led to patient records for hospitals in the wider area being affected [8]. There are 2 points to note from these events. Firstly, the extent of the damage caused in each of the incidents was difficult to

[1] A cascade may be defined as a causally related sequence of undesirable events. However, later in this paper, we will use a definition of cascade that does not require the events to be causally related.

E. Rome and R. Bloomfield (Eds.): CRITIS 2009, LNCS 6027, pp. 201–212, 2010.

predict at the time. Certainly, had the cascades' occurrence and evolution been better predicted (or detected earlier), preventative and mitigation measures might have limited the consequences of the cascade. Such uncertainty is characteristic of many cascade events in CIs and suggests that CI dependencies, and there implications, are not yet well understood. Secondly, in both examples there were dependencies present that exacerbated the cascades. Indeed, investigations undertaken after the cascades occurred exposed the role of a number of dependencies in facilitating the cascades. Via these dependencies (e.g. the geographic proximity of IT database systems to the fuel depot in the Buncefield incidence) the state of some CI component (e.g. explosion at Oil depot) was related to the state of some other component (e.g. database storing healthcare records), possibly in another CI. Therefore, a change in the number, or nature, of the dependencies in a CI may affect the occurrence, and size, of cascades.

In this paper we present an approach to modelling CIs, taking dependencies into account. Using the models we follow 2 lines of inquiry. Firstly, we study how the strength of dependencies affects the occurrence and size of cascades in the CIs. Upon varying the strength of dependencies we estimate, via Monte–Carlo simulation, the distributions of cascade sizes for the various CIs. A comparison of these distributions indicates which CI are affected by a change in the strength of the dependence. Secondly, we explore the question of what the consequences of a less detailed model are for modelling the occurrence and size of cascades. Certainly, due to the size and complexity of CIs, it is unreasonable to model "everything" in the real systems. Therefore, given some level of abstraction for the model how much benefit, if any, is gained by using a more detailed level of abstraction?

To illustrate the analysis approach we model interconnected CIs in the Rome area. The data and parameter values for the model are based on:

- a model of CIs in the Rome area developed within the IRRIIS[2] project [9] and inspired by a Telecommunications blackout that occurred in Rome [10-12];
- a *Preliminary Interdependency Analysis* (PIA) carried out to define and limit the scope of the model and to identify dependencies [10-12]. The model scope includes a specification of what the model's level of abstraction should be, which entities should be modelled explicitly, and what the state–spaces for the modelled entities are. The identified dependencies are used to define, in part, how modelled components are correlated. Such data is necessary for a mechanism we use to take dependencies into account in the stochastic models of CI;
- failure and repair rate field–data for Power and Telecommunication network components and equipment. The data was provided by SIEMENS and Telecom Italia [12];
- realistic parameter values for power network components including voltage levels, thermal limit capacities, and line impedances. This data was provided by SIEMENS;
- a compilation of real–life data on thousands of cascades from all over the world. This cascade data was compiled over several years by TNO (*Netherlands Organization for Applied Scientific Research*) [12].

[2] Integrated Risk Reduction of Information–based Infrastructure Systems (IRRIIS) is an EU project concerned with developing both a platform for simulating CI and technologies for mitigating against the negative consequences of CI interdependencies.

The outline of the paper is as follows. In Section 2 we discuss our approach to modelling CI. Section 3 outlines the implementation of simulations based on our models. Section 4 discusses the results of simulation-based studies we conducted while Section 5 summarizes conclusions and details directions of further work.

2 Stochastic Modelling of Critical Infrastructures

The stochastic models of CI we build are *Stochastic/Random processes* that are, themselves, made up of dependent *Stochastic/Random processes*. The examples given in Section 1 hint at uncertainty associated with predicting the occurrence and size of cascades. In our model such uncertainty in cascades results from uncertainty in what the next state of each component is, given the current state of the model. Each component in each CI is modelled as a *Random process*. These Random processes, as a consequence of identified dependencies (correlations) within and between the CIs, are probabilistically dependent. So, the interconnected CIs are, themselves, modelled as dependent Random processes. Also, there are factors external to the CIs, such as weather or terrorist attacks, which are modelled as Random processes that interact with the other Random processes. These external, environmental entities can put a strain on the CI, significantly affecting CI operation. So, the dynamical behaviour of the CIs is modelled as a Random process consisting of all of these aforementioned interacting, dependent, Random processes.

To define each Random process we define both a state–space and probabilities that govern the transitions of the component from state to state. In particular, for every distinct pair of states, i and j, we define conditional, *instantaneous transition rates*. These rates parameterize probability distributions used in a *competing risk*s model to determine what the potential next state for a component is. They also determine the potential *sojourn time* for the component; that is, how long before the component enters into its potential next state. For example, in the present work the components have the states $\{OK,Failed\}$. As a consequence of the components' state–spaces there are 2 instantaneous transition rates associated with each component; *a failure and repair rate*.

The existence of dependencies between components justifies correlation between the components. CI components can be said to be dependent if correlation exists between their states and/or state transitions. While there are other definitions of dependence in use this definition allows us to model a wide class of phenomena. All the dependencies that we have come across in the literature [13, 14] imply correlations. We implement this notion in our models as follows. Given that a component changes state the dependencies specify which other components will have their state–transition behaviour affected. So, there is a notion of *parent* (a component whose state changes affect the stochastic behaviour of some other components) and *child* (a component whose stochastic behaviour is affected by state changes in some other node) components. Each component has 2 related sets; the set of all of its parents and the set of all of its children. Parents can be children and children can be parents. In addition, the parent child relationship can be cyclical so parents can be children of their children (or children can be parents of their parents). By specifying all such parent–child relationships in the model we define a *Directed–Graph* representing all of the pairs of correlated components. We refer to this graph as "*the graph of stochastic associations for the model*". Whenever a parent component undergoes a state change the *failure and repair rates* of

its child components take on values that are conditional on the current states of all the components parents. This mechanism is *the primary way in which we model dependence*. The occurrence and effects of various dependence inducing phenomena may be modelled this way, including the consequences of human operator actions, natural disasters, geographic proximity terrorism and weather, on CI operation.

In addition to the primary mechanism for modelling dependence there are certain parent components whose state changes *deterministically* affect the states of some of its children. For example, the failure of a power component could result in the overloading of power lines in the power network. In some of our models which power lines are overloaded is determined by using a "*Linearized, DC, load flow approximation*" [6] to calculate real power flow across the power network. A static load profile (i.e. the consumption and supply of power remains unchanged) is used, as a first step, in our modelling. Another example of modelled deterministic consequences can be found in the Telecommunications network. Each Telecommunications node has a primary power source (power from a local Power distribution company) and a secondary power source (generators or batteries). Given that both the primary and secondary power sources are in failed states the Telecommunications nodes will, with certainty be in an inoperable state.

Component failures may result in failure cascades. Failure cascades can have different causes, may occur over different time–scales, might involve different components and could have different consequences, depending on which network the cascade occurs in. For instance, a sudden surge of power may result in a cascade of power line trips due to power line overloading. The effects of this sort of cascade can be seen almost immediately; some parts of the power network may lose power. Compare this with the aforementioned Buncefield explosion (see Section 1) which caused hospital records to be affected, long after the explosion occurred. Arguably, this cascade is quite different from the previous cascade. However, note two characteristics common to both cascade examples. Firstly, there is an interval of time within which at least one component is in a failed state. Secondly, within this interval of time there is some time point, t, at which there is a maximum number of simultaneously failed components. This suggests the following definition for cascade. Any maximal[3], uninterrupted time interval continuously throughout which at least one component of the relevant sub-network (e.g. Telco Network, Power distribution Network, etc.) is in a failed state defines a cascade of that network. Over any such time interval the maximal number of simultaneously failed components is the cascade–size. So, by definition, the size of any cascade is an integer ≥ 1.[4] Certainly, this definition of cascade–size is not unique and may not be the preferred definition of cascade for all situations. For instance, it may be more interesting to "weight" nodes in terms of functional importance, location in the network or economic impact of disruption. Our models allow for the cascade–size to be defined in terms of such alternative approaches, using the "*rewards*" functionality provided by the tool, *Möbius* [15, 16].

[3] In the sense that both the start point and the end point of that interval is either a start or endpoint of the simulation, or is an endpoint outside which the number of failed components is zero.

[4] While this definition implies that a cascade may be "trivial" (the failure of a single node would be defined as a cascade) this is simply a classification choice we have made largely for ease of presentation.

3 Simulating Critical Infrastructure

The CI models were used to estimate cascade–size distributions. This is achieved using the *Möbius* tool [15-17]. In particular, the model is created in *Möbius* using the *Stochastic Activity Networks* formalism[5]. This allows us to simulate Continuous–time, Discrete state–space, Random processes using event–driven, Monte–Carlo simulation. Three interacting CIs in Rome were modelled. However, we discuss the results for only the *Telecommunications network* and the *Power distribution network*. The *Power Transmission* network is the 3rd modelled network. We also discuss the results for the aggregated system comprising of all 3 CIs. We refer to this as the "Rome Power–Telco System" or "the entire model". Two studies were conducted using Möbius-based simulations of the Rome Power-Telco System. One study looks into the effect of the "*strength of dependence*" on the distribution of cascade-sizes, while the other study compares modes at different "*levels of abstraction*". Each study consists of a comparison between 2 experiments; a "*base-level*" experiment and a "*comparison*" experiment. The "*base-level*" experiment is the experiment that has been calibrated using all the data sources at our disposal. Consequently, it is the starting point for any comparison experiments as these are only "slight" modifications of the base–level experiment. For the "*strength of dependence*" study the comparison experiment will have parameter values almost identical to the base experiment except that the strength of some dependence is set at a noticeably different level. For the "*levels of abstraction*" study the comparison experiment uses an alternative, less sophisticated algorithm for determining line trips in the power network. So, 3 experiments in total were conducted. Each experiment simulates 10^5 hours of operation (just over 11 years and 4 months) in each of at least 15,000 simulation replications from which sample-mean, cascade–size occurrence rates were obtained. Experiment 1 is the base-level experiment, against which the other experiments will be compared. Experiment 2 changes the strength of certain dependencies while keeping the mean number of cascades in the power distribution network approximately the same as Experiment 1. Experiment 3 substitutes the "*Linearized, DC, load – flow approximation*", used in Experiment 1, for a simple algorithm that governs line trips in the power distribution network. The mean number of cascades in the whole model is kept approximately the same as that for experiment 1.

There are 3 model parameters relevant for the studies. The *Conditional Power-substation Failure rate coefficient* is the amount by which the failure rates of Medium and High voltage substations are scaled when the substations have lost communication with the *Supervisory Control and Data-Acquisition* (SCADA) system. So, this parameter is used to alter the strength of Power components dependence on Telco components for communication. Similarly, the *Conditional Telco-component failure rate coefficient* is the amount by which the failure rate of Telco nodes that have lost their primary source of power supply, but still have a secondary power source, is scaled. So, this parameter is used to model the strength of Telco components' dependence on power sources for their operation. Finally, the *Conditional probability of power-line overloading* is the conditional probability of a given power-line

[5] Stochastic Activity Networks are a generalisation of Stochastic Petri–nets.

overloading, given the failure of some other power component in the Power Network. Thisparameter is used in experiments that do not use the *Linearized, DC, load flow approximation* to determine power–line overloading.

4 Discussion of Simulation Results

For the base–level experiment the *Conditional Power-substation Failure rate* coefficient has a value of 10^3 and the *Conditional Telco-component failure rate* coefficient has a value of 10^5. Notice, from Fig. 1, that about 10 cascades of size greater than 4 occur in the combined Telco–Power system. However, the Power distribution network has about 8.87×10^{-3} of its cascades having a size greater than 4. So, the Power distribution network appears to contribute relatively little to the cascade size for the larger cascades. In contrast, there are about 2.53 cascades of size greater than 4 that occur in the Telco network; two orders of magnitude more than the related number for the Power distribution network. While this does not fully account for the number of cascades greater than 4 in the combined model it suggests that a significant number of relatively large cascades occur in other parts of the Rome model not depicted, i.e. the Power transmission network.

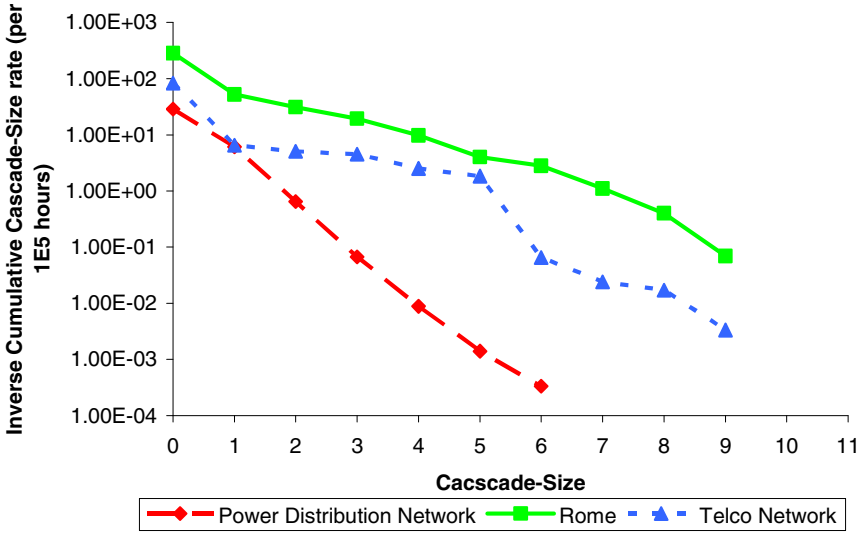


Fig. 1. For the base–level experiment this graph depicts the sample–mean number of cascades of size strictly greater than the indicated size (on the horizontal axis) that occur in each sub-network. For instance, the mean number of cascades that occurred in each of the sub–networks is given by the respective value of each graph corresponding to the cascade–size value of 0. So, the mean number of cascade events in the Power distribution network is 28.72, the mean number of cascade events in the entire model is 285.12 and the mean number of cascade events in the Telco network is 83.18.

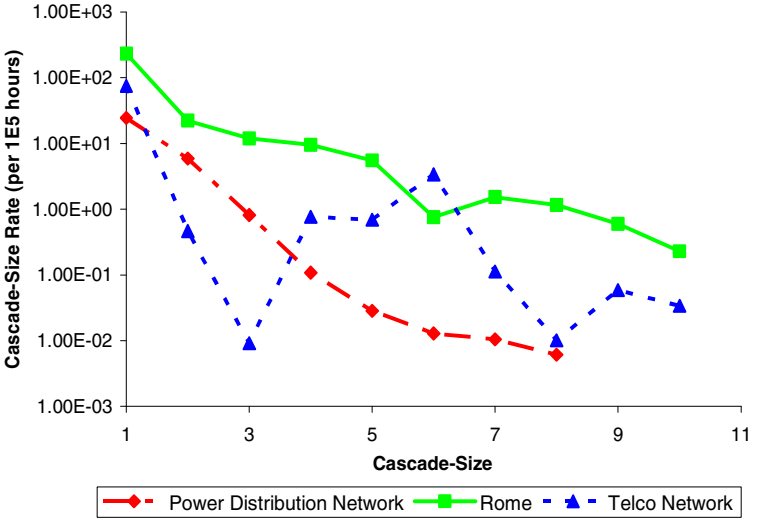


Fig. 2. Cascade–size distribution for each sub–network in the "Strength of dependence" experiment

The "Strength of dependence" experiment explores the dependence of Power components on Telco components and vice–versa. The value for the *Conditional Power-substation Failure rate coefficient* is set to 4800, which is higher than the corresponding value of 1000 in the base–level experiment. So, each power sub–station's dependence on Telco–services in this experiment is almost 5 times stronger than in the base–level experiment. Contrastingly, the *Conditional Telco-component failure rate coefficient* parameter is set to 1.0 for this experiment, where it had a value of 10^5 in the previous experiment. So, each Telco component experiences a dependence on Power services that is *5 orders of magnitude weaker* than it was in the previous experiment. The DC approximation to AC power flow is still used. The resulting Cascade–size distribution is given in Fig. 2. The parameter values were chosen so that the mean number of cascades in the entire model, 286, is comparable with the respective value in the base–level experiment, 285.

The Power distribution network still contributes relatively little to the large cascades. There is no noticeable change, even though a change exists, in the contribution of Power distribution cascades to the total number of cascades in the model. The Telco network appears to exhibit a rapid drop from a frequency of 75.1 for single failures to a frequency of 9.13×10^{-3} for triple failures. So, there were hardly any cascades of size 3 among the cascades in the Telco network. However, this steep fall is followed by a steep rise so that there are an estimated 3.4 cascades of size 6 occurring; *a change in cascade–size rate of 2 orders of magnitude* between cascade–size 2 and cascade–size 6. While in both experiments a significant number of cascades of size 6 occur in the Telco network it would seem that the effect of having weaker dependence on Power–services in the current experiment is to reduce the number of

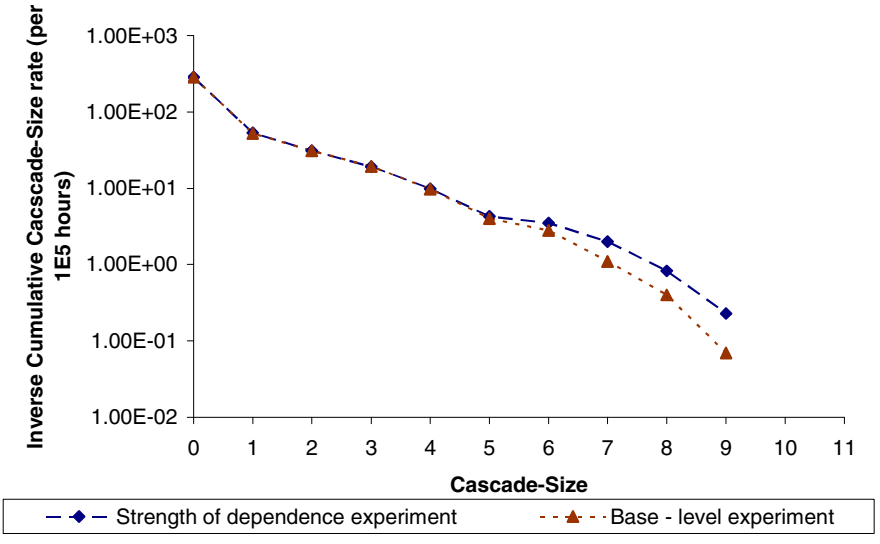


Fig. 3. Comparison of the Inverse, Cumulative Cascade–size rates for the Rome model

cascades of size 3. The cascade–size distributions for the entire model are virtually identical, up until cascades greater than size 5 (see Fig. 3). Only in the extreme right of the graph are the curves an order of magnitude apart. So, the Telco nodes' having 5 orders of magnitude less of a dependence on Power components has little effect, globally, on the distribution of cascades if compensated by a comparatively modest increase in the dependence of Power nodes on Telco services. In Fig. 3 even though the graphs are arguably similar there is, nevertheless, a strict ordering between the graphs for "large" cascades (cascades with sizes greater than 6). "Large" cascades are strictly less likely to occur in the base–level experiment. Similar orderings are visible and more pronounced for the cascade–size distributions in both the Telco network and the Power distribution network. In particular, consider the Power distribution network (see Fig. 4), where the ordering is maintained and, additionally, the distributions appear to diverge. One of the distributions is fairly linear on a log-linear plot and the other has approximately 2 linear regimes (one between cascade sizes 1 and 3 and the other between 4 and 6). This suggests 3 exponential laws governing the cascade-size distributions, with one of the distributions having 2 distinct laws characterizing it. Statistical tests of significance and estimating the parameters for these laws is of immediate interest. In summary, globally there appears to be little effect on cascade – size distributions from a weaker dependence of Telco components on Power services and stronger dependence of Power nodes on Telco services. Locally, however, there are differences between the distributions.

In the "levels of abstraction" experiment we use a relatively naïve algorithm for power–line overloading on the cascade–size distributions, replacing the *Linearized DC approximation to AC power flow*. A Binomial probability distribution randomly trips power–lines, given the failure of a power component. For the occurrence of

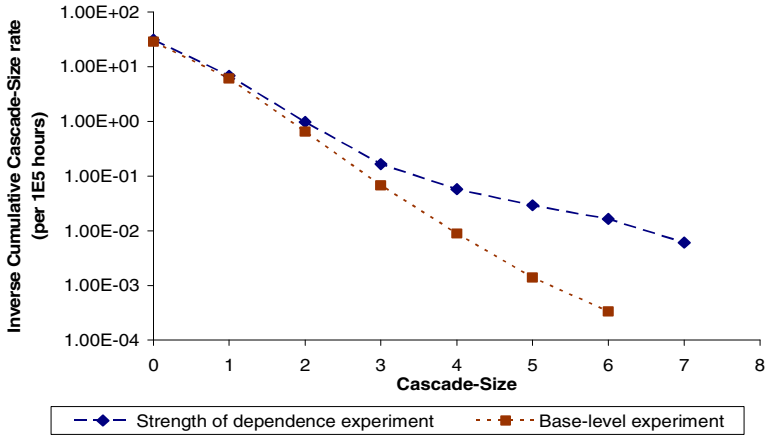


Fig. 4. Comparison of the Inverse, Cumulative, Cascade–size rates for the Power distribution network

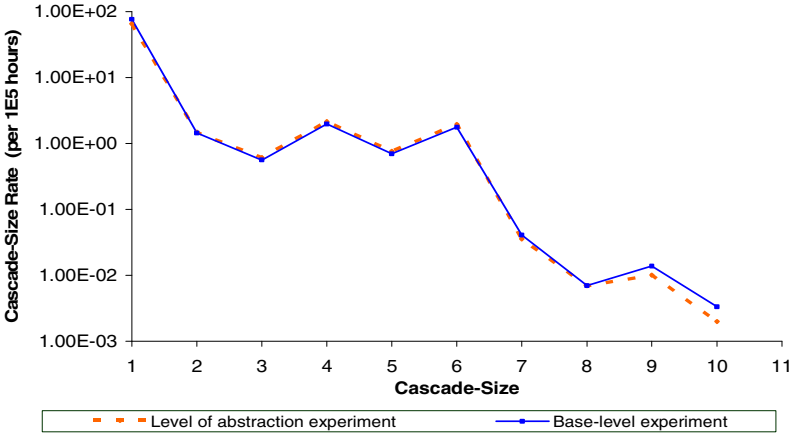


Fig. 5. Comparison of Cascade – size distributions for the Telecommunications network

cascades in each network will it matter that we are using an extremely simplistic algorithm to model power–line trips? If it does, how significant is the change in model behaviour? The value of the parameter *Conditional probability of power-line overloading*, which we set as 3.2×10^{-3} for this experiment, defines the Binomial distribution. All other parameters are the same as the base–level experiment. The mean number of cascade events in the Power distribution network is $28\cdot48$, which is comparable with $28\cdot72$ for the base–level experiment. The distributions for the Telco network are visibly almost identical (see Fig. 5). On the other hand, the power distribution network exhibits significant differences. A clear ordering exists, with the

base–level experiment having an order of magnitude more occurrences of "large" cascades. The shape of the distributions both appear to be approximately linear and parallel on a log-linear plot. Thus, again the data suggests an exponential relationship characterising the cascade size-distributions.

In conclusion, we note that while a significant change in strength of dependence noticeably affected the Cascade-size distributions for the sub-networks, the impact on the whole model was pronounced only for very large cascades. Also, the use of an extremely simplistic algorithm for power line trips had a significant impact in the power distribution network but was negligible in the Telco network.

5 Conclusion and Future Work

This paper presents a quantitative approach to modelling interdependencies in CIs and results from applying the approach to a non-trivial case-study, the Rome Power-Telco system developed within the EU IP IRRIIS.

We argue in favour of using probabilistic models of interdependencies and provide details of the model of 'stochastic associations' we developed. Such models not only allow the user to incorporate any knowledge that might exist about the likelihood and dependence between various adverse events (e.g. failures of the modelled component) and risks of environmental disruptions (e.g. natural disasters, extreme weather conditions, etc.) but also to study the impact and challenge various assumptions about these.

The results presented target primarily the impact of the level of abstraction on the modelling results, an important scalability issue with very large CIs. We report, that in some cases the behaviour of 'low fidelity' models is close to the behaviour of more sophisticated models (i.e. low fidelity models can offer an acceptable accuracy). This result seems important, because it opens up a practical way of modelling very large CIs with a reasonable accuracy.

In the light of these results we propose a future modelling approach in which a network of CIs is decomposed into manageable parts. Comparative studies (between high and low fidelity models, as described above) of these parts may then be applied so that the low-fidelity models can be tuned to model accurately the behaviour of the modelled sub-systems. Then, from the perspective of each CI, the network of CI may be modelled by combining acceptable low-fidelity models of some parts with high fidelity models of other parts. Although such an approach seems plausible further work is needed to validate that the thus composed model will still be accurate. We intend to attack this issue in our future work.

In addition, the work demonstrates that a range of parameter values can give similar model behaviour, depending on what aspects of the model are of interest. Very different parameter values gave the same total number of cascades in the model. However, the *tails* of the cascade-size distributions for the whole model and the power distribution network exhibited divergence. This suggests that in parameterizing such models for use in practice care must be taken since the effect of different parameter values is to produce "regimes of agreement" between the models. In the case we observed the models may diverge for large cascades but agree reasonably well for small cascades. The magnitude of this effect differed between networks.

In the current work, as a first approximation, static load profiles were used in the study. Dynamic load profiles that capture daily, or seasonal, variations in load may have the effect of significantly changing the probability of large-scale cascades. We will experiment with, and study the effects of, dynamic load profiles as an extension of the current work.

A related problem is addressing aspects of 'observability' of the state of the entire CI. The Rome system consists of networks, operated by organisations which may have detailed knowledge of their own network but very limited knowledge of the state of the networks operated by other operators. Approaches for dealing with such limited observability are of practical interest. We have scoped in [12] an approach to on-line risk estimation (RE) based on the probabilistic models described in this paper. Validating RE in terms of achievable accuracy of risk predictions (i.e. whether the periods with predicted high risk of disruption will indeed tend to be highly positively correlated with actual disruptions) is a problem, which we are currently working on.

In addition to the relative confidence intervals used for determining acceptable convergence of the estimated distribution data points further statistical tests of significance will be carried out on the data to increase confidence in the results. Also, the data suggests a number of exponential relationships governing some of the cascade-size distributions. The parameters for these laws will be estimated as part of future work.

Acknowledgements

This work is supported by the EU IP IRRIIS, Contract Number 027568.

References

1. Bloomfield, R., et al.: Report on Service Oriented Interdependency Analysis (IRRIIS Project Deliverable D2.2.4). City University London, London (2007)
2. Bloomfield, R., et al.: Infrastructure interdependency analysis: an introductory research review. Adelard, London (2009)
3. Boccaletti, S., et al.: Complex networks: Structure and dynamics. Physics Reports 424, 175–308 (2006)
4. Reka, A., Barabasi, A.L.: Statistical mechanics of complex networks. Reviews of Modern Physics, 74 (2002)
5. Pederson, P., et al.: Critical Infrastructure Interdependency Modeling: A Survey of U.S. and International Research (2006)
6. Schläpfer, M., Kessler, T., Kröger, W.: Reliability Analysis of Electric Power Systems Using an Object-oriented Hybrid Modeling Approach. In: Proceedings of the 16th Power Systems Computation Conference, Glasgow (2008)
7. U.S.-Canada Power System Outage Task Force. Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations (2003), http://reports.energy.gov/
8. Board, B.M.I.I.: The Buncefield Incident December 11, 2005 The final report of the Major Incident Investigation Board (2008), http://www.buncefieldinvestigation.gov.uk/index.htm
9. IRRIIS. Integrated Risk Reduction of Information-based Infrastructure Systems (EU project, 2006–2009)

10. IRRIIS deliverable D2.2.4., Report On Service Oriented Interdependency Analysis (2007), http://www.irriis.org
11. IRRIIS deliverable D2.2.2., Tools and techniques for interdependency analysis (2007), http://www.irriis.org/File.aspx?lang=2&oiid=9138&pid=572
12. IRRIIS deliverable D2.2.6, Preliminary Interdependency Analysis (PIA) as a service-oriented approach towards LCCI Interdependency Analysis (report and prototype) (2009)
13. Rinaldi, S.M., Peerenboom, J.P., Kelly, T.K.: Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies. IEEE Control Systems Magazine 21(6), 11–25 (2001)
14. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, Palo Alto (1988)
15. Sanders, W.H., et al.: The Mobius Manual (2008)
16. Sanders, W.H., Meyer, J.F.: Stochastic Activity Networks: Formal Definitions and Concepts. In: Lectures on Formal Methods and Performance Analysis, Berg en Dal, The Netherlands. First EEF/Euro Summer School on Trends in Computer Science. Springer, Berlin (2001)
17. Courtney, T., et al.: The Möbius Modeling Environment. In: Tools of the 2003 Illinois International Multiconference on Measurement, Modelling, and Evaluation of Computer-Communication Systems (2003)

Author Index