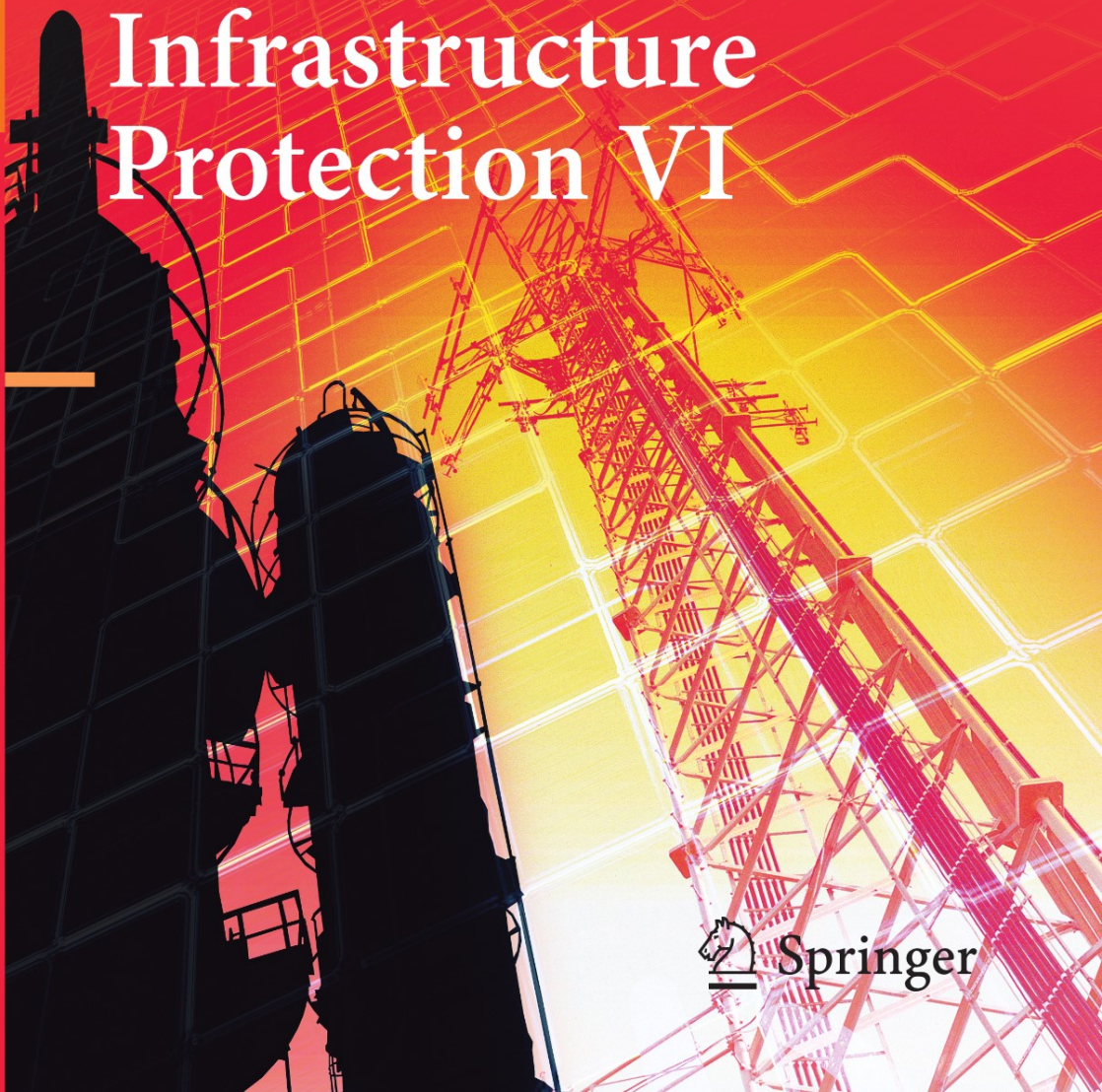
IFIP AICT 390

Jonathan Butts
Sujeet Shenoi
(Eds.)

Critical Infrastructure Protection VI



Springer

IFIP Advances in Information and Communication Technology 390

IFIP – The International Federation for Information Processing

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP's aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

> *IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.*

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP's events range from an international congress to local seminars, but the most important are:

- The IFIP World Computer Congress, held every second year;
- Open conferences;
- Working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is also rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is about information processing may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.

Jonathan Butts Sujeet Shenoi (Eds.)

Critical Infrastructure Protection VI

6th IFIP WG 11.10 International Conference
ICCIP 2012
Washington, DC, USA, March 19-21, 2012
Revised Selected Papers

Springer

Volume Editors

Jonathan Butts
Air Force Institute of Technology
Wright-Patterson Air Force Base
Dayton, OH 45433-7765, USA
E-mail: jonathan.butts@afit.edu

Sujeet Shenoi
University of Tulsa
Tulsa, OK 74104-3189, USA
E-mail: sujeet@utulsa.edu

ISSN 1868-4238 e-ISSN 1868-422X
ISBN 978-3-642-35763-3 e-ISBN 978-3-642-35764-0
DOI 10.1007/978-3-642-35764-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012954250

CR Subject Classification (1998): K.6.5, D.4.6, K.5.1, K.4.1, I.6.3-5, C.2.0, H.4.2-3, H.3.4-5, K.6.1

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

Contributing Authors ix

Preface xv

PART I THEMES AND ISSUES

1
The European Perspective of Telecommunications as a Critical Infrastructure 3
Fabio Bisogni, Simona Cavallini, Luisa Franchina, and Giovanni Saja

2
Implementing Critical Information Infrastructure Protection Structures in Developing Countries 17
Ian Ellefsen and Sebastiaan von Solms

3
Integrity-Organization Based Access Control for Critical Infrastructure Systems 31
Abdeljebar Ameziane El Hassani, Anas Abou El Kalam, and Abdellah Ait Ouahman

PART II CONTROL SYSTEMS SECURITY

4
Analysis of Field Devices Used in Industrial Control Systems 45
John Mulder, Moses Schwartz, Michael Berg, Jonathan Van Houten, Jorge Mario Urrea, and Alex Pease

5
A Firmware Verification Tool for Programmable Logic Controllers 59
Lucille McMinn and Jonathan Butts

6
Quantifying Controller Resilience Using Behavior Characterization 71
Henry Bushey, Juan Lopez, and Jonathan Butts

7
Using Bloom Filters to Ensure Access Control and Authentication Requirements for SCADA Field Devices 85
Jeffrey Hieb, Jacob Schreiver, and James Graham

8
Agent Interaction and State Determination in SCADA Systems 99
Thomas McEvoy and Stephen Wolthusen

PART III INFRASTRUCTURE SECURITY

9
Infrastructure Protection in the Dutch Financial Sector 113
Matthijs van Oers, Leon Strous, and Ron Berndsen

10
Privacy-Preserving Power Usage Control in the Smart Grid 127
Chun Hu, Wei Jiang, and Bruce McMillin

11
Effects of Time Delays in the Electric Power Grid 139
Hasan Ali and Dipankar Dasgupta

12
Measuring Name System Health 155
Emiliano Casalicchio, Marco Caselli, Alessio Coletta, Salvatore Di Blasi, and Igor Nai Fovino

13
Emergency Messages in the Commercial Mobile Alert System 171
Paul Ngo and Duminda Wijesekera

PART IV INFRASTRUCTURE MODELING AND SIMULATION

14
A One-Dimensional Sparse Space-Time Specification of the Generalized Railroad Crossing 187
Michael Gosnell and Bruce McMillin

15
A Networked Evidence Theory Framework for Critical Infrastructure Modeling 205
Chiara Foglietta, Andrea Gasparri, and Stefano Panzieri

16
Enabling the Exploration of Operating Procedures in Critical Infrastructures 217
Christos Siaterlis, Bela Genge, Marc Hohenadel, and Marco Del Pra

Contributing Authors

Anas Abou El Kalam is a Professor of Computer and Network Security at the National School of Applied Sciences, Marrakesh, Morocco. His research interests include embedded systems security, network security, wireless security, security models and intrusion detection.

Abdellah Ait Ouahman is a Professor of Telecommunication Networks and the Director of the National School of Applied Sciences, Marrakesh, Morocco. His research interests include logistics, telecommunications and information networks.

Hasan Ali is an Assistant Professor of Electrical and Computer Engineering at the University of Memphis, Memphis, Tennessee. His research interests include advanced power systems, smart grid and micro grid systems, renewable energy systems, energy storage systems and flexible AC transmission systems.

Abdeljebar Ameziane El Hassani is a Ph.D. student in Network Security Science at Cadi Ayyad University, Marrakesh, Morocco, and at the National Polytechnic Institute, Toulouse, France. His research interests are in the area of access control for distributed critical infrastructures.

Michael Berg is a Principal Member of the Technical Staff at Sandia National Laboratories, Albuquerque, New Mexico. His research interests include optimization and embedded systems design.

Ron Berndsen is the Head of the Oversight Department at The Netherlands Bank, Amsterdam, The Netherlands; and an Endowed Professor of Financial Infrastructure and Systemic Risk at the University of Tilburg, Tilburg, The Netherlands. His research interests are in the area of financial market infrastructures.

Fabio Bisogni is a Member of the Board of the Formit Foundation, Rome, Italy. His research interests include critical infrastructure protection, cyber security, critical event management and information disclosure policy.

Henry Bushey is an M.S. student in Cyber Operations at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. His research interests include SCADA systems security and protocol verification.

Jonathan Butts, Chair, IFIP Working Group 11.10 on Critical Infrastructure Protection, is an Assistant Professor of Computer Science at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. His research interests include critical infrastructure protection and cyber physical systems security.

Emiliano Casalicchio is a Researcher in the Department of Computer Science, University of Rome – Tor Vergata, Rome, Italy; and a Senior Advisor at the Global Cyber Security Center, Rome, Italy. His research interests include performance-oriented design and evaluation of large-scale distributed systems, and analysis, modeling and simulation of critical information infrastructures.

Marco Caselli is a Ph.D. student in Computer Science at the University of Twente, Enschede, The Netherlands. His research interests are mainly in the field of cyber security applied to industrial infrastructures.

Simona Cavallini is a Senior Researcher at the Formit Foundation, Rome, Italy. Her research interests include critical infrastructure protection, interdependency analysis, economics of security and macroeconomics modeling.

Alessio Coletta is a Researcher at the Global Cyber Security Center, Rome, Italy. His research interests include critical infrastructure protection, industrial control systems security and malware analysis.

Dipankar Dasgupta is a Professor of Computer Science at the University of Memphis, Memphis, Tennessee. His research interests include evolutionary and immunological computation, intrusion detection and fault detection.

Marco Del Pra is a Software Designer with Stratiqo, Milan, Italy; and an External Consultant at the Institute for the Protection and Security of the Citizen, Joint Research Centre of the European Commission, Ispra, Italy. His research interests include the modeling and simulation of industrial applications, and the development of applied mathematics and real-time software.

Salvatore Di Blasi is a Researcher at the Global Cyber Security Center, Rome, Italy. His research interests are in the area of information and communications systems security.

Ian Ellefsen is a Senior Lecturer in the Academy of Computer Science and Software Engineering at the University of Johannesburg, Johannesburg, South Africa. His research interests include critical infrastructure protection and critical information infrastructure protection models for developing nations.

Chiara Foglietta is a Ph.D. student in Computer Science and Automation at the University of Roma Tre, Rome, Italy. Her research interests include data fusion techniques, situational awareness and the application of multi-agent systems to critical infrastructure protection, especially power systems and smart grid security.

Luisa Franchina is the Director General of the Critical Infrastructure Secretariat in the Office of the Military Advisor to the Italian Presidency of the Council of the Ministers, Rome, Italy. Her research interests include security and business continuity.

Andrea Gasparri is an Assistant Professor of Computer Science and Automation at the University of Roma Tre, Rome, Italy. His research interests include mobile robotics, sensor networks and networked multi-agent systems.

Bela Genge is a Post-Doctoral Researcher at the Institute for the Protection and Security of the Citizen, Joint Research Centre of the European Commission, Ispra, Italy. His research interests include critical infrastructure protection, intrusion detection systems, and the security and resilience of networked industrial control systems.

Michael Gosnell is a Ph.D. student in Computer Science at Missouri University of Science and Technology, Rolla, Missouri. His research interests include fault detection, tolerance, prognostics and diagnostics, parallel and distributed computing, and wireless and unmanned systems.

James Graham is the Henry Vogt Professor of Computer Science and Engineering at the University of Louisville, Louisville, Kentucky. His research interests include information security, digital forensics, critical infrastructure protection, high performance computing and intelligent systems.

Jeffrey Hieb is an Assistant Professor of Engineering Fundamentals at the University of Louisville, Louisville, Kentucky. His research interests include information security, honeypots, digital forensics, secure operating systems and the use of technology in engineering education.

Marc Hohenadel is an Action Leader at the Institute for the Protection and Security of the Citizen, Joint Research Centre of the European Commission, Ispra, Italy. His research interests include the security of networked critical infrastructures.

Chun Hu is a Ph.D. student in Computer Science at Missouri University of Science and Technology, Rolla, Missouri. His research interests include privacy-preserving data mining and information assurance.

Wei Jiang is an Assistant Professor of Computer Science at Missouri University of Science and Technology, Rolla, Missouri. His research interests include privacy-preserving data mining, data integration, text sanitization and applied cryptography.

Juan Lopez is a Research Engineer with the Center for Cyberspace Research at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. His research interests include critical infrastructure protection and radio frequency identification.

Thomas McEvoy is a Ph.D. student in Mathematics at Royal Holloway, University of London, London, United Kingdom; and a Technical Manager at HP Information Security, Bracknell, United Kingdom. His research interests include the modeling and simulation of critical infrastructures and hybrid systems in relation to security properties.

Bruce McMillin is a Professor of Computer Science at Missouri University of Science and Technology, Rolla, Missouri. His research interests include critical infrastructure protection, computer security, formal methods, distributed systems and parallel algorithms.

Lucille McMinn is an M.S. student in Computer Science at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio. Her research interests include malware analysis and embedded device security.

John Mulder is a Senior Member of the Technical Staff at Sandia National Laboratories, Albuquerque, New Mexico. His research interests include computer networks and industrial control systems security.

Igor Nai Fovino is the Head of the Research Division at the Global Cyber Security Center, Rome, Italy. His research interests include critical infrastructure protection, intrusion detection, secure communication protocols and industrial informatics.

Paul Ngo is a Ph.D. candidate in Computer Science at George Mason University, Fairfax, Virginia; and the Next Generation Network (NGN) Security Lead at the National Communications System in Arlington, Virginia. His research interests are in the area of emergency communications systems.

Stefano Panzieri is an Associate Professor of Computer Science and Automation, and the Director of the Automation Laboratory at the University of Roma Tre, Rome, Italy. His research interests include industrial control systems, robotics and sensor fusion.

Alex Pease is a Member of the Technical Staff at Sandia National Laboratories, Albuquerque, New Mexico. His research interests include memory and kernel security and integrity.

Giovanni Saja is a Security Manager at Telecom Italia, Milan, Italy. His research interests are in the area of critical infrastructure protection.

Jacob Schreiver is an M.S. student in Computer Engineering at the University of Louisville, Louisville, Kentucky. His interests include computer security, image processing and the use of computer games in education.

Moses Schwartz is a Member of the Technical Staff at Sandia National Laboratories, Albuquerque, New Mexico. His research interests include supply chain risk management, visualization, formal methods and embedded systems.

Christos Siaterlis is a Scientific Officer at the Institute for the Protection and Security of the Citizen, Joint Research Centre of the European Commission, Ispra, Italy. His research interests include various aspects of the security, stability and resilience of complex systems, especially critical infrastructures such as the Internet and smart grid.

Leon Strous, IFIP President, is an IT Auditor in the Cash and Payment Systems Division of The Netherlands Bank, Amsterdam, The Netherlands. His research interests include business continuity, operational crisis management and critical infrastructure protection, with special emphasis on the financial sector.

Jorge Mario Urrea is a Senior Member of the Technical Staff at Sandia National Laboratories, Albuquerque, New Mexico. His research interests include information assurance, wireless networks and microcontroller development.

Jonathan Van Houten is a Senior Member of the Technical Staff at Sandia National Laboratories, Albuquerque, New Mexico. His research interests include digital design and embedded software security.

Matthijs van Oers is a Senior Policy Advisor with The Netherlands Bank, Amsterdam, The Netherlands. His research interests are in the broad area of financial infrastructure protection, in particular payment and securities, and business continuity and crisis management.

Sebastiaan von Solms is a Research Professor in the Academy of Computer Science and Software Engineering at the University of Johannesburg, Johannesburg, South Africa. His research interests include information security and critical information infrastructure protection.

Duminda Wijesekera is an Associate Professor of Information and Software Engineering at George Mason University, Fairfax, Virginia. His research interests include information, network, telecommunications and control systems security.

Stephen Wolthusen is a Professor of Information Security at the Norwegian Information Security Laboratory, Gjovik University College, Gjovik, Norway; and a Reader in Mathematics at Royal Holloway, University of London, London, United Kingdom. His research interests include critical infrastructure modeling and simulation, and network and distributed systems security.

Preface

The information infrastructure – comprising computers, embedded devices, networks and software systems – is vital to operations in every sector: information technology, telecommunications, energy, banking and finance, transportation systems, chemicals, agriculture and food, defense industrial base, public health and health care, national monuments and icons, drinking water and water treatment systems, commercial facilities, dams, emergency services, commercial nuclear reactors, materials and waste, postal and shipping, and government facilities. Global business and industry, governments, indeed society itself, cannot function if major components of the critical information infrastructure are degraded, disabled or destroyed.

This book, *Critical Infrastructure Protection VI*, is the sixth volume in the annual series produced by IFIP Working Group 11.10 on Critical Infrastructure Protection, an active international community of scientists, engineers, practitioners and policy makers dedicated to advancing research, development and implementation efforts related to critical infrastructure protection. The book presents original research results and innovative applications in the area of infrastructure protection. Also, it highlights the importance of weaving science, technology and policy in crafting sophisticated, yet practical, solutions that will help secure information, computer and network assets in the various critical infrastructure sectors.

This volume contains sixteen edited papers from the Sixth Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection, held at the National Defense University, Washington, DC, March 19–21, 2012. The papers were refereed by members of IFIP Working Group 11.10 and other internationally-recognized experts in critical infrastructure protection.

The chapters are organized into four sections: themes and issues, control systems security, infrastructure security, and infrastructure modeling and simulation. The coverage of topics showcases the richness and vitality of the discipline, and offers promising avenues for future research in critical infrastructure protection.

This book is the result of the combined efforts of several individuals and organizations. In particular, we thank Robert Miller, Heather Drinan, Nicole Hall Hewett and Firoozeh Rahimian for their tireless work on behalf of IFIP

Working Group 11.10. We gratefully acknowledge the Institute for Information Infrastructure Protection (I3P), managed by Dartmouth College, for supporting IFIP Working Group 11.10. We also thank the Department of Homeland Security and the National Security Agency for their support of IFIP Working Group 11.10 and its activities. Finally, we wish to note that all opinions, findings, conclusions and recommendations in the chapters of this book are those of the authors and do not necessarily reflect the views of their employers or funding agencies.

Jonathan Butts and Sujeet Shenoi

I

THEMES AND ISSUES

Chapter 1

THE EUROPEAN PERSPECTIVE OF TELECOMMUNICATIONS AS A CRITICAL INFRASTRUCTURE

Fabio Bisogni, Simona Cavallini, Luisa Franchina, and Giovanni Saja

Abstract This paper attempts to analyze the degree to which the telecommunications sector is regarded as a critical infrastructure at the European level. Taking into account a new categorization of telecommunications applications and infrastructure perspectives, a new matrix-based classification method is proposed to clarify the protection approaches of policy makers and telecommunications asset owners and operators. The so-called "criticality matrix" approach applied to the Italian environment demonstrates the different perspectives held by policy makers and telecommunications asset owners and operators, and shows how all the stakeholders may engage a common base to define efficient and effective strategies that can enhance the security and resilience of critical infrastructure assets.

Keywords: Europe, critical infrastructures, telecommunications, classification

1. Introduction

The importance of telecommunications to all the societal sectors has led European policy makers to include it in the list of potential critical infrastructures. This view is also corroborated by research on interdependencies between critical infrastructures and the consequent cascading effects of telecommunications failures (see, e.g., [1, 16–18]). Interested readers are referred to Luiijf, *et al.* [15] for an analysis of the interdependencies involving the telecommunications sector and other European critical infrastructure sectors.

In June 2004, the European Council requested that a comprehensive strategy be created for protecting critical infrastructures. In October 2004, the European Commission adopted a communication on critical infrastructure protection as part of the fight against terrorism [5]; this communication included several proposals for enhancing prevention, preparedness and response at the

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 3–15, 2012.

European level in the event of terrorist attacks on critical infrastructure assets. In December 2004, the Council, in its conclusions on prevention, preparedness and response to terrorist attacks, approved the European Commission's proposal to establish a European Programme for Critical Infrastructure Protection (EPCIP) that would spearhead various initiatives aimed at enhancing critical infrastructure protection.

The next year, in November 2005, the European Commission adopted a "green paper" [7] that outlined strategic alternatives with regard to critical infrastructure protection. The process culminated with a 2008 European Council directive that emphasized the identification and designation of European critical infrastructures and an assessment of the need to improve their protection [12].

Recognizing the existence of several infrastructures whose disruption or destruction could have a significant impact on member states, the European Council directive focused on clarifying the key elements related to critical infrastructure protection. In particular, the directive defined the following key concepts:

- **Critical Infrastructure:** Assets, systems or parts thereof located in European Union member states, which are essential for the maintenance of vital social functions, security, safety, health and economic/social welfare of the population, and whose destruction or malfunction would have a significant impact in a member state (loss of service).

- **European Critical Infrastructure:** Critical infrastructures located in European Union member states whose destruction or malfunction would have a significant impact in at least two European Union member states. The significance of the impact is to be assessed in terms of cross-cutting criteria, including the effects of cross-sector dependencies on other infrastructures.

The directive also defined a common approach for identifying European critical infrastructures and protecting them. Since the various sectors have accumulated extensive expertise and experience with regard to asset protection, the directive was designed to be implemented on a sector basis. At this time, two areas – energy and transportation and their related sub-sectors – are identified by the directive as domains in which the procedures for identifying European critical infrastructures must be applied.

The implementation of the directive has imposed a number of requirements on member states that impact the activities and organization of owners and operators of the identified European critical infrastructure assets. In particular, the identification of a European critical infrastructure on the part of a member state leads to a procedure with two requirements for asset owners and operators:

- **Implementation of Operator Security Plans:** The directive provides an indication of the minimum components that must be addressed in a plan. In particular, a plan should identify the critical infrastructure

assets and the security solutions that are in place and those that are being implemented. Also, the procedures should cover, at the very least: the identification of critical assets; a risk analysis that includes threats, vulnerabilities and potential impacts; the identification, selection and prioritization of countermeasures, categorized as those that are permanent and those that are enforceable gradually.

- **Appointment of Liaison Officers:** The directive requires liaison officers to act as a points of contact between the critical infrastructures and the national bodies responsible for their protection.

The 2008 directive also recognized the future need to expand the list of critical infrastructures. Indeed, it gave priority to the information and communications technology sector during the first review, which started in January 2012.

Following the 2008 directive and its goal of increasing the scope of the European critical infrastructure sectors, the European Commission issued a 2009 communication to protect Europe from large-scale cyber attacks and disruptions, and enhance preparedness, security and resilience [9]. This communication articulated European policy on strengthening security and trust in the information society. Focusing attention on prevention, preparedness and awareness, it specified immediate actions to strengthen the security and resilience of critical information and communication infrastructures, including all aspects of telecommunications services. Subsequent European Union debate spurred efforts to examine the challenges and priorities for network and information security policy and to set up the most appropriate instruments needed at the European level to ensure the security and resilience of critical information nfrastructures.

The 2009 communication on critical information infrastructures came at the end of a process extending back to 2005 that focused on the increasing role of the telecommunications sector in Europe. A 2005 Commission communication [6] highlighted the urgent need to coordinate efforts to build the trust and confidence of stakeholders in electronic communications and electronic services. To this end, a strategy for a more secure information society was adopted in 2006 [8]. This 2006 communication was produced to revitalize the European Commission strategy set out in 2001. The main intent was to develop a dynamic, global strategy in Europe, based on a culture of security and founded on three pillars: dialogue, partnership and empowerment. As part of the partnership framework, the 2006 communication asked the European Network and Information Security Agency (ENISA) to develop a trusted partnership with member states and stakeholders to develop an appropriate data collection framework, including procedures and mechanisms to collect and analyze European-Union-wide data on security incidents and consumer confidence. Member states, the private sector and the research community are required to establish strategic partnerships to ensure the availability of data on the information and communications technology security industry and on market trends for products and services in the European Union (EU). Moreover, to improve the ability to

respond to network security threats, the European Commission asked ENISA to examine the feasibility of a European information sharing and alert system to articulate effective responses to current and emerging threats.

The 2006 communication sought to achieve the infrastructure identification and protection objectives by adopting a multi-stakeholder approach and promoting effective public policy and private sector initiatives. According to this goal, in order to identify the key assets in the telecommunications sector and to adequately protect them, the relevance of the perspectives adopted by European and national policy makers should match the organizational approach of telecommunications operators in meeting their security and resilience requirements. Indeed, what is needed is a common framework in terms of approaches and goals that would support efficient and effective security and resilience strategies.

2. European Telecommunications Sector

The first step in developing a common framework for policy makers and asset owners and operators is to define the components of the telecommunications sector regardless of the geographic areas in which they are located. At present, statistics on the evolution of telecommunications markets are becoming wide and consistent, thanks to the efforts of several governmental and non-governmental organizations.

The availability of public data related to the larger information and communications technology sector is relatively broad. The primary reliable sources for the European context are Eurostat, Organization for Economic Cooperation and Development (OECD) and International Telecommunications Union (ITU). These entities conduct electronic and postal surveys as well as interviews at the national and international levels in order to obtain high-quality statistical data. The resulting data can be used to define indicators that are comparable across countries. However, limited data is available about telecommunications applications and usage.

Using the traditional classification of telecommunications applications in terms of fixed telephony, mobile telephony and Internet, the first point of reference is Eurostat, which, through the national statistical institutes, provides an annual dataset on the twenty-seven EU member states (EU-27). Information and communications technology market features are clarified through statistics that link telecommunications and various economic indicators of the member states.

A valuable Eurostat data asset is the elaboration of several aggregate indicators such as "Internet activities of enterprises." However, as far as telephony is concerned, data is provided in terms of the volume of different types of calls along with indicators related to operators and service providers. Eurostat data sources are also useful for analyzing information and communications technology use by citizens. Indeed, various statistics are available about the expanding role of information and communications technology in daily activities, especially related to "households" and "individuals." Several indicators pertaining

to network access also give a good sense of the importance of telecommunications services.

OECD databases and statistics are also valuable sources of information about the telecommunications sector. The available data includes communication channel access (fixed, mobile and Internet), employees, revenue, capital and investment. The main limitations of OECD data related to the European telecommunications sector are that they cover only part of the EU-27 and that most OECD statistics are published in books instead of being disseminated free of charge on the Internet. Nevertheless, the OECD Directorate for Science Technology and Industry frequently publishes some information and communications technology indicators from its various databases. These indicators cover trade, firms, use and growth related to information and communications technologies.

ITU, the U.N. telecommunications agency, maintains a highly reliable and often quoted database that contains data provided directly by governments via ITU's annual questionnaires. Data related to the telecommunications sector, which is available on a payment basis, cover more than 200 countries, including all the EU-27 countries from 1960 onwards. The data related to fixed and mobile telephony includes coverage, diffusion, traffic, prices, revenues and investments. The same data is also available for the entire telecommunications sector with the addition of some data about faults. In the case of the Internet, the available data primarily focuses on the numbers of users and subscribers.

Another reliable, but limited, data source is the annual report prepared by the European Telecommunications Network Operators Association (ETNO). The report provides data about the European telecommunications market for fixed telephony, mobile telephony and Internet, focusing mainly on operator revenue trends and ETNO member investments.

3. Infrastructure Approach

At the European level, telecommunications is considered to be a complex critical infrastructure sector. Also, national approaches and related regulations hinder a common approach – in the member states, the criteria for identifying critical infrastructures are defined by various *ad hoc* public institutions and government entities.

Most European member states use the following approaches when defining and identifying critical infrastructures:

- **Service-Oriented Approach:** This approach focuses on the services and/or functions that are vital to society. Infrastructures that provide these services and/or functions are considered to be critical infrastructures.

- **Asset-Oriented Approach:** This approach focuses on the impact and/or risk. Infrastructures whose disruption may result in major impact in terms of casualties, economic and public effects are considered to be critical infrastructures.

- **Operator-Oriented Approach:** This approach focuses on infrastructure operators because of their decision-making roles in providing vital services. Operators are considered critical on the basis of legislative obligations and spontaneous interactions because they help ensure the protection of assets and the resilience of services.

A 2011 study by IABG, Booz and Alcatel-Lucent [13] investigated the sectoral criteria for identifying European critical infrastructures. The study involved sixteen European countries and incorporated contributions by 68 organizations, mainly national entities with information and communications technology responsibilities and telecommunications operators. According to the study, most of the member states are considering the information and communications sector as outlined in Directive 2008/114/EC [12]. About two-thirds of the member states adopt the service-oriented approach (or its variations) in designating critical infrastructures, while just a few countries engage the asset-oriented approach or the operator-oriented approach. In some limited cases, critical infrastructures are identified based on common aspects of all three approaches.

4. Proposed Classification Method

The aforementioned 2009 European Commission communication [9] encourages member states to continue to develop, in cooperation with all relevant stakeholders, criteria for identifying critical infrastructure assets in the information and communications technology sector. According to many experts, the main obstacle to applying the criteria promoted by the European critical infrastructure directive [12] is the nature of the information and communications technology sector and the telecommunications component. Information and communications technology has a strategic role because it pervades all societal activities, but its horizontal nature prevents precise boundaries from being defined for the sector.

For this reason and others, a classification method for identifying the criticality of the telecommunications infrastructure should rely on a multi-faceted framework that is shared among the main stakeholders of the sector. Because of the quality of service and the security and resilience provisions, two dimensions must be considered: (i) telecommunications applications; and (ii) the adopted infrastructure approach.

The first dimension of the proposed classification method is telecommunications applications. The traditional classification of telecommunications applications as fixed telephony, mobile telephony and Internet is gradually losing its significance as a result of service convergence and communication channel integration. Although the main statistics in the telecommunications sector are collected using the traditional classification, the evolution of technologies as connected software, hardware and middleware suggests that a content-oriented taxonomy would be more appropriate. Therefore, our proposed method classifies telecommunications applications in terms of voice communications, data

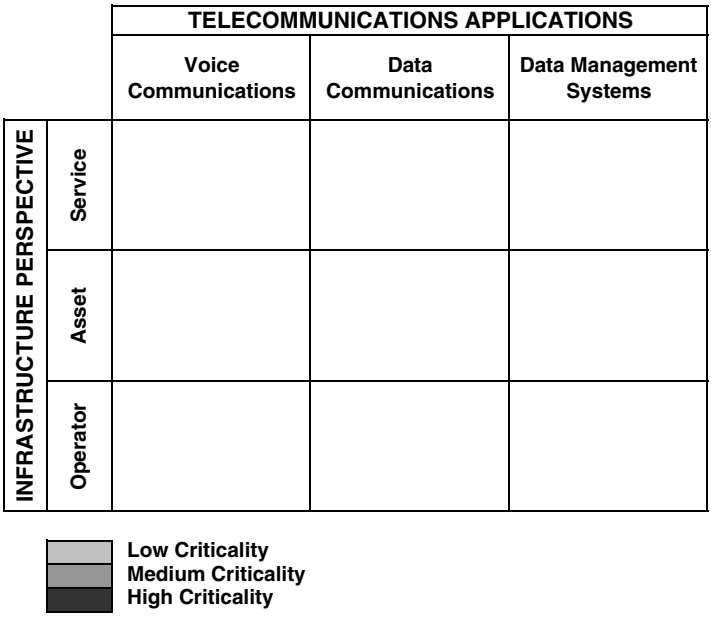


Figure 1. Classification matrix.

communications and data management systems (i.e., data processing, hosting and related activities).

The second dimension of the proposed classification method is the adopted infrastructure approach, which may be service-oriented, asset-oriented or operator-oriented. According to this dimension, the provided services (e.g., fixed telephony), essential assets (e.g., public switched telephone network facilities) and operators (e.g., Telecom Italia) are all infrastructure objects with security and resilience provisions.

The classification matrix presented in Figure 1 allows the various telecommunications sector stakeholders to define relative criticality levels (low-level, medium level and high-level) based on the appropriate thresholds. The matrix can be used to compare the frames of reference used by policy makers (e.g., national authorities) and owners and operators of potential critical information infrastructure assets (e.g., telecommunications operators) when defining policies and strategies for security and resilience.

5. Italian Case Study

This section presents a case study where the classification methodology described in the previous section is applied to the Italian telecommunications sector.

Table 1. Total expenses per operator in Italy in 2009 and 2010 [14].

Operator	2009	2010
Telecom Italia	**51.6%**	**48.9%**
Vodafone Italia	20.6%	21.4%
Wind	12.5%	13.6%
Fastweb	4.6%	4.9%
H3G	3.7%	4.2%
BT Italia	2.8%	2.7%
Others	4.3%	4.4%
Total	**100.0%**	**100.0%**

5.1 Italian Telecommunications Sector

According to the implementation of Article 53 of the Italian Legislative Decree on the Electronic Communication Code (CCE) of August 1, 2003, all users within the national boundaries, regardless of their geographical locations, should be able to access "universal communication services" at a pre-defined quality level. In addition, telecommunications companies must provide these universal communication services to all users at an affordable price.

At present, Telecom Italia is Italy's only unique provider of universal communications services. According to Article 54 of CCE, Telecom Italia must provide fixed telephony services in addition to free emergency services. With regard to security and resilience, specific quality targets for fixed telephony services related to the line disruption rate and recovery time were initially regulated by Article 61, but have been updated in recent years. These indications match the findings of the 2011 study by IABG, Booz and Alcatel-Lucent [13], which noted that Italy tends to uses the service-oriented approach for identifying critical information infrastructures.

Telecom Italia's position as the leading Italian telecommunications provider has historical roots. Telecom Italia was established in 1994 as result of the merger of STET and SIP, which was the only Italian telecommunications company since 1964. It was not until 1997 that the Italian telecommunications market was opened to other national and international operators in the fixed telephony, mobile telephony and Internet sub-sectors. In 2010, at least six operators provided both fixed and mobile telephony to the Italian market. However, according to the latest (2010) data in Table 1, Telecom Italia is the largest in terms of expenses for fixed and mobile telephony (48.9%). The other main players are Vodafone Italia (21.4%) and Wind (13.6%).

Using a sub-sector classification similar to the traditional classification, the market shares of fixed telephony connections and bandwidth connections also show the dominance of Telecom Italia (Table 2). However, Vodafone Italia is the principal provider of mobile telephony services.

Table 2. Market shares of Italian telecom operators in 2010 [14].

Operator	Fixed Connections[1]	Bandwidth Connections[1]	Mobile Voice and Data
Telecom Italia	**71.6%**	**53.9%**	35.7%
Vodafone Italia	7.4%	12.0%	**36.8%**
Wind	10.6%	14.8%	18.4%
Fastweb	7.5%	12.9%	NA[2]
H3G	NA[2]	NA[2]	7.2%
BT Italia	0.4%	0.9%	NA[2]
Tiscali	1.9%	4.1%	NA[2]
Others[3]	1.4%	13.6%	1.9%
Total	**100.0%**	**100.0%**	**100.0%**

[1] December 2010 data.
[2] Operator does not provide the service or its market share is included in Others.
[3] Includes mobile virtual network operators for mobile telephony.

5.2 Telecom Italia

Telecom Italia is the largest Italian provider of fixed telephony services. It is a major international player with 57,853 employees in Italy and a total of 84,335 people worldwide (according to the 2011 corporate report). In 2010, the company had industrial investments of 4,583 million euros, turnover of 27,571 million euros and earnings before interest, taxes, depreciation and amortization (EBITDA) of 11,412 million euros.

The domestic (Italian) infrastructure of Telecom Italia includes 31.3 million mobile telephony lines (Telecom Italia Mobile), 9.1 million broadband connections (1.9 million of them wholesale costumers) and 15 million retail fixed line connections (7.2 million of them broadband connections).

The international operations of Telecom Italia are primarily focused on South America: 55.5 million mobile lines in Brazil (25.5% share); 17.4 million mobile lines, 4.1 million fixed lines and 1.5 million broadband connections in Argentina; and 2 million mobile lines in Paraguay.

With regard to Telecom Italia, the definition of infrastructure criticality is essential in order to apply the mandated security and resilience measures. As the main Italian company, Telecom Italia is regarded by the Italian Government as the owner and operator of a potential critical infrastructure, and the operator-oriented approach is applicable because of Telecom Italia's role as the sole provider of universal communications services.

However, in order to guarantee the protection of the telecommunications infrastructure, a strong asset-oriented perspective is adopted internally by Telecom Italia. Assets are defined as those that are physical and substantial. The criticality of the assets is determined by a three-step process: defining the asset boundaries, assessing the quality of the provided services and guaranteeing the

level of user satisfaction. Taking into account these three connected aspects, the criticality depends on the weakest link in the chain. Thus, a critical infrastructure can be considered to be an asset that makes a service available, and the absence of the service produces significant disruptions for end users. For this reason, the criticality lies in "transporting" the service to end users.

For security and resilience purposes, identifying assets and the related processes of service production and delivery are crucial. The optimal solution is a homothetic organization. Each structure in such an organization has its own responsibilities, updates the risk map, sets up operator security plans, defines recovery and continuity plans, and guarantees the compliance of the implemented activities in order to avoid functional overlap.

The Telecom Italia approach involves defining the asset boundaries and assessing the criticality of the assets. Producing an exhaustive and unambiguous list of assets supports the identification of the key elements that are required to deliver services (business processes, technological functions, human resources, facilities, etc.). The vulnerabilities and related risks are evaluated for each asset by applying heuristic techniques (e.g., a quasi-logarithmic scale) because the probabilities of occurrence of rare and high-impact critical events are usually not available. This technique, which is based on a non-linear scale, ensures more precise evaluations because it highlights the exceptional occurrence of minimum and maximum values.

The computation of the expected impact as a score for each location in terms of vulnerability multiplied by risk allows a ranking of the criticality of the locations themselves. A logarithmic ranking of locations and their assets is then created according to different levels of criticality. The logarithmic ranking assists in separating critical assets from less strategic assets. Having a smaller number of critical assets also helps focus protection efforts.

Based on this approach, Figure 2 shows the criticality matrix classification results for Telecom Italia. From the service perspective, voice communications, which is related to fixed telephony and has special regulatory attention, is not ranked as the application with the highest criticality. However, from the asset-oriented perspective, data management systems has the highest criticality.

The proposed classification method has two main benefits. The first benefit is the opportunity to increase stakeholder awareness about the criticality of the telecommunications infrastructure according to different perspectives while also enabling the comparison of infrastructure criticality. In fact, during the course of the case study, Telecom Italia had to reason about its perspective with regard to managing criticalities and about the perspectives adopted by other stakeholders, including policy makers and competitors.

The second benefit is policy making support. The criticality matrix summarizes certain structural features of the telecommunications framework such as the influence of legal provisions (e.g., regarding the universal communications services provided by Telecom Italia) and the direct liability of the telecommunications infrastructure (e.g., ownership of the physical infrastructure, especially fixed telephony).

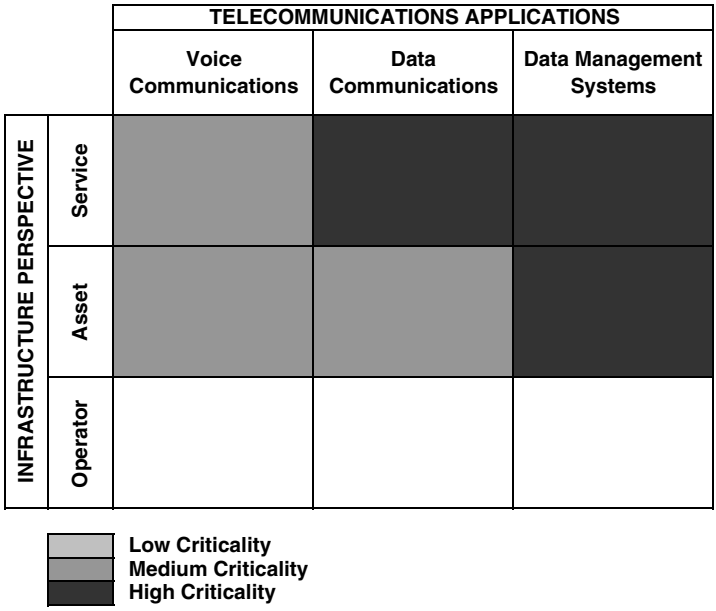


Figure 2. Classification method applied to Telecom Italia.

Thus, the criticality matrix method provides a homogeneous classification. In particular, it enables the various stakeholders to assess the criticality of telecommunications applications according to the specific infrastructure perspectives they adopt. Note, however, that in order to enhance the utility of the classification method, it is recommended to collect and use data based on the three types of telecommunications applications shown in Figure 2.

6. Conclusions

The growing societal reliance on the telecommunications sector has made it imperative to ensure that the critical information and communications infrastructure is secure and resilient. While there is broad agreement on the need to protect critical infrastructures, different perspectives and approaches are applied at the European level to identify – and ultimately protect – infrastructure assets. This heterogeneity has driven the effort to develop the criticality matrix classification method. The method offers a common, shared framework for policy makers and asset owners and operators to identify key infrastructure assets and to assess their criticality, helping formulate effective and efficient security and resilience strategies. The Telecom Italia case study demonstrates the different perspectives and decision processes that can be applied to identify infrastructure assets and assess their criticality; and how the criticality matrix method can be used to provide a highly homogeneous classification that

also facilitates the comparison of results obtained using different infrastructure perspectives.

References

[1] F. Bisogni and S. Cavallini, Assessing the economic loss and social impact of information system breakdowns, in *Critical Infrastructure Protection IV*, T. Moore and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 185–198, 2010.

[2] F. Bisogni, S. Cavallini, R. Bellotti, M. Tancioni, L. Remotti, A. Wright, G. Gasperini and S. Anselmucci, The Vulnerability of Information Systems and its Inter-Sectoral, Economic and Social Impacts – VIS, Final Project Report, Formit Foundation, Rome, Italy, 2009.

[3] S. Cavallini, S. Di Trocchio, F. Bisogni, M. Tancioni and P. Trucco, Development of a Methodology and Research of Quantitative Data on the Economics of Security and Resilience in Critical Communications and Information Infrastructures – SMART-SEC, Final Project Report, Formit Foundation, Rome, Italy, 2010.

[4] European Commission, Network and Information Security: Proposal for a European Policy Approach, Commission Communication COM(2001) 298, Brussels, Belgium, 2001.

[5] European Commission, Critical Infrastructure Protection in the Fight Against Terrorism, Commission Communication COM(2004) 702 Final, Brussels, Belgium, 2004.

[6] European Commission, A European Information Society for Growth and Employment, Commission Communication COM(2005) 229 Final, Brussels, Belgium, 2005.

[7] European Commission, Green Paper on a European Programme for Critical Infrastructure Protection, Commission Communication COM(2005) 576 Final, Brussels, Belgium, 2005.

[8] European Commission, A Strategy for a Secure Information Society – Dialogue, Partnership and Empowerment, Commission Communication COM(2006) 251, Brussels, Belgium, 2006.

[9] European Commission, Protecting Europe from Large-Scale Cyber Attacks and Disruptions: Enhancing Preparedness, Security and Resilience, Commission Communication COM(2009)149 Final, Brussels, Belgium, 2009.

[10] European Commission, A Digital Agenda for Europe, Commission Communication COM(2010) 245, Brussels, Belgium, 2010.

[11] European Parliament and Council, On a Common Regulatory Framework for Electronic Communications Networks and Services (Framework Directive), Directive 2002/21/EC, Brussels, Belgium, 2002.

[12] European Parliament and Council, On the Identification and Designation of European Critical Infrastructures and the Assessment of the Need to Improve Their Protection, Directive 2008/114/EC Brussels, Belgium, 2008.

[13] IABG, Booz and Alcatel-Lucent, Study to Support the Process to Define Sectoral Criteria to Identify European Critical Infrastructures in the ICT Sector, with Particular Focus on the Sub-Sectors of Internet, Fixed and Mobile Telecommunications, Final Project Report, Ottobrunn, Germany, 2011.

[14] Italian Authority for the Protection of Communications (AGCOM), Relazione Annuale 2011 sull'Attivita svolta e sui Programmi di Lavoro (in Italian), Naples, Italy (`www.agcom.it/Default.aspx?message=downloadpdf&DocID=131`), 2011.

[15] E. Luiijf, A. Nieuwenhuijs, M. Klaver, M. van Eeten and E. Cruz, Empirical findings on critical infrastructure dependencies in Europe, *Proceedings of the Third International Workshop on Critical Information Infrastructure Security*, pp. 302–310, 2009.

[16] P. Pederson, D. Dudenhoeffer, S. Hartley and M. Permann, Critical Infrastructure Interdependency Modeling: A Survey of U.S. and International Research, Technical Report INL/EXT-06-11464, Idaho National Laboratory, Idaho Falls, Idaho, 2006.

[17] J. Santos and Y. Haimes, Modeling the demand reduction input-output inoperability due to terrorism of interconnected infrastructures, *Risk Analysis*, vol. 24(6), pp. 1437–1451, 2004.

[18] J. Sarriegi, F. Sveen, J. Torres and J. Gonzalez, Adaptation of modeling paradigms to the critical infrastructure interdependencies problem, *Proceedings of the Third International Workshop on Critical Information Infrastructure Security*, pp. 295–301, 2009.

Chapter 2

IMPLEMENTING CRITICAL INFORMATION INFRASTRUCTURE PROTECTION STRUCTURES IN DEVELOPING COUNTRIES

Ian Ellefsen and Sebastiaan von Solms

Abstract The development of a national critical information infrastructure protection (CIIP) structure is essential to safeguard critical systems from cyber attacks and other threats. As developing nations leverage Internet technologies, it is imperative that they develop their own national CIIP structures to ensure reliable operations, incident response and resilience in the face of attacks. This paper presents a framework designed to enable developing countries to define a set of clear deliverables that can be used to realize a national CIIP structure.

Keywords: Critical information infrastructure protection, developing countries

1. Introduction

Technologically advanced countries have implemented a variety of critical information infrastructure protection (CIIP) structures to safeguard their national information infrastructures and critical systems from cyber attacks and other threats. Historically, developing nations have had poor access to Internet-based technologies [1], which has limited their need to develop effective CIIP structures such as computer security incident response teams (CSIRTs) [12].

However, this situation is changing. Many developing nations are experiencing massive growth in Internet capacity and the use of Internet-based technologies. Attacks on the information infrastructure can severely affect the ability of a country to function effectively [16]. If commercial entities were to lose Internet services for a prolonged period, the economic effects would be significant. The impact of large-scale cyber attacks on national critical systems would be much more devastating. It is clear that developing countries are finding them-

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 17–29, 2012.

selves in the situation where they have to implement national CIIP structures to safeguard their information infrastructure assets [2].

This paper presents a framework that is intended to be used by developing countries to implement CIIP structures. To this end, the paper investigates the role of traditional CIIP mechanisms such as CSIRTs and related protection structures. The generic framework outlines a set of deliverables that allow for the establishment of a national CIIP structure.

2. Background

Developing countries are making massive investments in Internet and communications technologies. Many large-scale infrastructure assets used for electricity distribution, water supply, and banking and finance are utilizing these technologies to improve their ability to deliver services. The resulting information infrastructure is transforming the manner in which governments interact with citizens, companies transact business, and individuals access vital information and services.

Despite their reliance on the information infrastructure, developing countries rarely implement a nationally-coordinated protection structure to protect their vital information assets [5]. Cyber attacks, such as distributed denial of service (DDoS) attacks, can severely affect all the infrastructure sectors [3]. Cyber attacks differ greatly from traditional types of attacks. Historically, the ability to wage war has been the domain of governments. However, cyber attacks can potentially be initiated by any person with relatively little expenditure and without the need for a high degree of technical proficiency [15], and these attacks can have a direct effect on all sections of society.

In the United States, 85% of all critical systems are owned and operated by private entities [18]. The situation is quite different in most developing countries, where the majority of infrastructure assets are in the public sector. But regardless of the extent of government ownership, there should be a transition from centralized information infrastructure protection structures to public structures that safeguard commercial and individual interests. This is not to suggest that there is no place for governmental structures, only that they should operate in tandem with public protection structures.

3. Protection Structures

There are a number of protection structures that can form the basis of a national CIIP structure. The structures, which are intended to provide a coordinated platform for dealing with cyber incidents, are geared towards the specific environments in which they are deployed. However, despite their differences, protection structures take one of two forms, a top-down structure or bottom-up structure. We discuss a computer security incident response team (CSIRT) as an example of a top-down structure, and a community-oriented security, warning and advice (C-SAW) team as an example of a bottom-up structure.

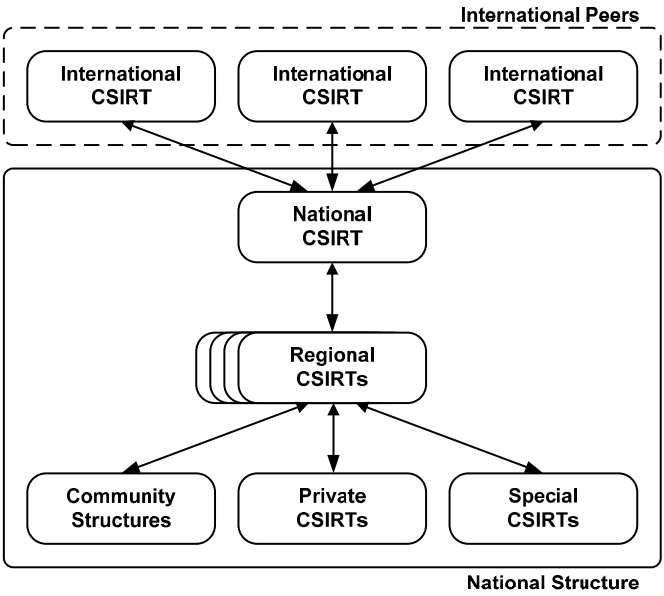


Figure 1. High-level CSIRT model.

Computer Security Incident Response Team. A CSIRT is a national entity that coordinates proactive and reactive efforts focused on managing cyber security incidents [8]. Figure 1 presents the different components of a CSIRT structure, which provide incident handling support to various constituencies.

The CSIRT itself follows a top-down model, where coordination is provided at the national level, with a number of regional CSIRTs that provide support to smaller constituencies [14]. Each CSIRT is structured to suit the environment in which it is deployed [19].

The primary service provided by a CSIRT is incident response. Incident response covers a number of services that seek to identify, manage and mitigate cyber security threats [4, 19].

A CSIRT is national in its scope and, as such, maintains relationships with international peers, governments and large organizations. However, the needs of individuals and small organizations cannot be overlooked in a national CIIP structure. The needs of these segments of the population are addressed by computer security, advisory and warning (C-SAW) teams.

Community-Oriented Security, Advisory and Warning Teams. Small businesses and individual households make up a large percentage of computer users. Often these users have to fend for themselves when dealing with cyber security threats and incidents.

A community-oriented structure is required to enable these smaller stakeholders to receive cyber security support. This structure is a "bottom-up" model, where security support is provided in a loosely-coupled manner from within a community.

Community-oriented security, advisory and warning (C-SAW) teams are an example of a community-oriented model that could be deployed within a national CIIP structure. These teams provide CSIRT-like services to a smaller, less informed community of members [6, 7].

A C-SAW team can also serve as an intermediary between a larger national CIIP structure and the smaller stakeholders, with a direct focus on providing cyber security support to its community. A C-SAW team should be community driven and operated by members of the community it services [7]. The services provided by a team, which typically involve vulnerability tracking and incident response, are largely dictated by the needs of its community.

C-SAW teams should operate independently of the larger national CIIP structure. Nevertheless, a C-SAW team should maintain good communication channels with other teams as well as the national CIIP structure.

C-SAW teams are important to national CIIP efforts because small businesses and individuals may not have the technical expertise available to manage CIIP threats and incidents. These smaller stakeholders should not be ignored because incidents that affect large numbers of these users can severely affect critical infrastructure operations [6]. Other community-based structures serve a similar role as a C-SAW team. The common aspect is that they are community-driven and focus on providing support to their communities.

Overall Structure. The CSIRT and C-SAW teams should operate together to provide the front-end for a national CIIP structure. The design and operation of these teams are vital to ensure that CIIP efforts pervade all sections of society. However, the mere deployment of protection structures is not sufficient to establish a successful national CIIP structure. Indeed, a national CIIP structure typically goes through a number of developmental stages before it can provide adequate protection for a nation's information infrastructure.

4. CIIP Framework for Developing Nations

The International Telecommunications Union (ITU) has produced a number of documents related to the development of CIIP structures in developing countries. One of the key documents is the *ITU Cybersecurity Work Programme to Assist Developing Countries 2007-2009* [13], which highlights the need to establish effective CIIP structures in developing nations because of the potential impact that they have on the global economy. This section builds on the ITU effort by describing a framework for developing effective national CIIP structures in developing countries.

Cyber security structures in developed nations have their roots in the early years of the Internet, when the U.S. Defense Advanced Research Projects Agency (DARPA) funded the development of the initial Computer Emergency

Response Team (CERT) in response to the Morris worm [14]. These structures have evolved with the Internet to meet new and expanding requirements.

Developing nations, on the other hand, have historically had limited Internet access and poor provisioning of information and telecommunications infrastructures. However, the introduction of a number of high capacity fiber optic cables – especially in Sub-Saharan Africa [17] – has created a situation where the establishment of effective CIIP structures is a necessity. When designing and implementing these CIIP structures, developing nations have the advantage of being able to leverage the lessons learned from the efforts undertaken by developed countries.

Developing countries have unique challenges that should be addressed by CIIP structures. In particular, Harris [10] has identified the following key challenges:

- Rapid development of information infrastructures.
- High-levels of cyber security illiteracy.
- Significant use of mobile technologies.
- High demand to adopt and provision web services.
- Inadequate legislation addressing cyber security.
- Inadequate policy documentation addressing cyber security.

All these challenges must be addressed in a national cyber security policy. Of course, the scope with which the challenges are addressed would depend on the conditions and needs of the country in question.

It is also important that the CIIP structure provides support to all sections of society. Furthermore, it is necessary to consider the needs of the private sector that may own and operate a significant portion of the critical infrastructure, as well as small businesses and individuals that make up a large segment of computer users within a developing country [9].

5. Two-Factor Development

A national CIIP structure must provide cyber security support to two primary societal groups. The first group is served by a traditional CSIRT structure, and the second by community-based structures. Each group exhibits different cyber security needs, and the overall CIIP structure should be able to address these needs. Ideally, the development of a national CIIP structure should follow a "two-factor development" strategy, where protection for each societal group is developed in parallel, with the holistic structure developing over time.

Figure 2 shows the society groups and the CIIP structures that are responsible for providing cyber security support and managing incidents. We now discuss the roles of the CSIRT and C-SAW teams.

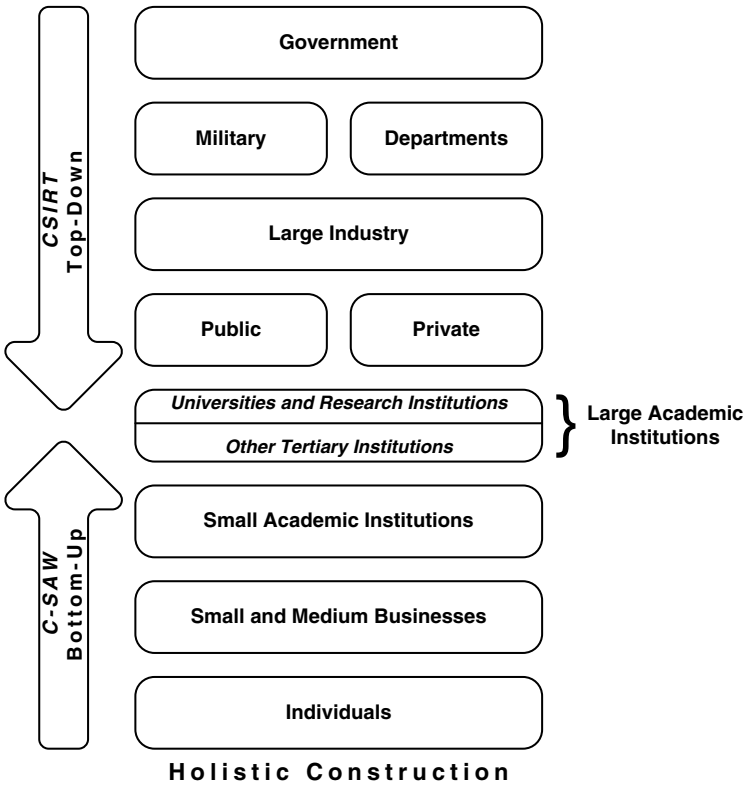


Figure 2. Organizations in a holistic CIIP structure.

CSIRT Role. The CSIRT is the lead entity in a national CIIP structure implemented according to the top-down model. The CSIRT coordinates the national cyber security policy, develops cyber security strategies and provides oversight and management for security related incidents in diverse operating environments. Figure 2 lists the specific organizations that fall under the direct management of a CSIRT structure. The organizations are:

- **Government Entities:** Governmental departments, military entities, and other government-sponsored utilities.
- **Large Industry Entities:** Financial institutions, telecommunication, manufacturing, electrical power, water distribution, sewage treatment and other large industries that provide services at a national level.
- **Large Academic Entities:** Large tertiary academic institutions and national research institutions.

The management of these organizations is delegated to a CSIRT because the organizations have large numbers of users and substantial computing resources. Also, the organizations may control critical infrastructure assets and may require considerable computing resources and network bandwidth. An additional benefit of assigning these entities to an emerging CSIRT structure is that they often have internal security structures and policies in place, which can be utilized to aid in the development of the CSIRT.

C-SAW Team Role. C-SAW teams are responsible for managing and coordinating cyber security efforts for smaller entities. These entities include:

- **Small Academic Entities:** Primary and secondary academic institutions.
- **Small Industry Entities:** Small and medium-sized businesses.
- **Individuals:** Private citizens.

These entities may be smaller than those managed by a CSIRT and they are often overlooked [11], but they are, nevertheless, vital to a national infrastructure. Early CIIP structures primarily focused on large industry entities. However, due to the abundance of individual users and small businesses and their potential to serve as breeding grounds for malware, these entities can have a serious impact on critical national systems.

Following the bottom-up paradigm, the services provided by a C-SAW team are driven by community needs. C-SAW teams play a vital role in educating their communities and helping secure their assets [6]. The teams also provide a bridge between their communities and the national CIIP structure.

6. CIIP Framework Development

The development of a national CIIP framework occurs in an incremental manner. The parallel development of top-down and bottom-up structures over the evolution of the national framework results in a holistic CIIP structure.

6.1 Stages of Development

A national CIIP structure progresses through several phases of development before it can be fully operational. Previous work related to the development of national CIIP structures (see, e.g., [14, 19]) does not translate well into the context of developing nations. This is largely due to the constraints imposed by the environment in developing countries. Therefore, it is necessary to conduct a rigorous assessment of the environment in which the CIIP structure will be deployed.

The following three phases of CIIP structure development are geared towards developing nations:

- **Initial Development:** Environmental assessments are conducted, legislation is evaluated, technological aspects are assessed and basic structures

are put in place. This phase sets the groundwork for the later stages of development.

- **Intermediate Development:** CIIP structures are developed to support growing needs. Community-based structures are expanded and public awareness schemes are initiated.

- **Mature Development:** CIIP structures are fully able to handle cyber security incidents.

6.2 Initial Development

The initial development phase is primarily concerned with laying the groundwork for the national CIIP structure. Many assessments are conducted during this phase, with the goal of deploying a functional CSIRT. The assessments are aimed at understanding the environment that must be protected, and identifying the strengths and weaknesses of existing systems.

Environmental Assessment. The goal of this assessment is to understanding the key components of the environment. Areas addressed in the environmental assessment include:

- **Critical Systems:** Identification of the set of nationally critical systems. The set of systems could be derived from assessments conducted in developed countries. These systems would eventually fall under the jurisdiction of the CSIRT.

- **Stakeholders:** Identification of the stakeholders in the CIIP structure. These are role players who have a vested interest in the stability of national systems. The stakeholders include government departments, large companies and international partners. The degree of interest of each stakeholder must also be gauged.

- **Legislation:** Identification of existing legislation related to cyber security and projected changes to the legislation.

- **Expertise:** Identification of the expertise required to develop the CIIP structure. This helps provide recommendations on whether a country has the local capacity to develop an effective national cyber security structure.

The role of the environmental assessment is to understand the environment where the CIIP structure will be deployed. The assessment is not limited to the components listed above. Indeed, due to the unique nature of each deployment environment, a number of other factors may have to be considered during the assessment.

Legislative Assessment. The legislative assessment is concerned with identifying the legal environment in which the CIIP structure is to be deployed.

As is often the case in developing countries, the legal system may not make provisions for current and future developments in technology. Despite the fact that legal frameworks are complex and diverse, the legislative assessment can be broken down into two basic components:

- **Current Legislation:** Evaluation of the current set of legislation that addresses cyber security, national cyber security policy, physical information infrastructure and compliance with international best practices.

- **Possible Amendments:** Identification of the areas of legislation that may have to be amended to allow for the effective deployment of a national CIIP structure. This component should consider the assignment of legal powers to the various entities in a CIIP structure to enable them to operate effectively.

Once again, these are not the only tasks that to be performed. Further analysis should be conducted during this phase to identify other legal issues.

Technology and Vulnerability Assessment. A technology and vulnerability assessment is conducted in order to gain an understanding of the operating environment. This assessment should identify technological components and their vulnerabilities that could potentially impact the national CIIP structure. The assessment should cover the following aspects:

- **Current and Future Bandwidth:** Investigations of the amount of available bandwidth, and well as future projections.

- **New Technologies:** Investigations of new technologies that could impact information infrastructure security. This would also include investigations of mobile technologies.

- **Current Systems:** Investigations of the current state of computer-based systems and their impact on overall cyber security. The analysis should also cover legacy systems.

In addition to helping understand the current operating environment, the technology and vulnerability assessment enables the national CIIP structure to accommodate longer-term projections.

International Peer and Partner Assessment. The role of international partnerships in CIIP cannot be overlooked. Partnerships with international CIIP structures can provide valuable assistance in creating effective local structures in developing countries. The partnerships are also important during the later phases of development. Due to the international nature of the Internet, these partnerships cannot be ignored. Incident information must be shared freely between international peers to identify and mitigate the effects of security incidents.

Local entities are a valuable resource during the initial phase of CIIP structure development. These entities include multinational companies and large local companies that already have cyber security structures in place.

A key component is fostering trust between international and local partners. Active participation in international organizations such as the Forum of Incident Response and Security Teams (FIRST) can help build trust.

Small-Scale Deployment. After the deployment environment has been assessed, a small-scale deployment of a CIIP structure (CSIRT or C-SAW team) should be attempted to identify any oversights in the initial assessments. The trial deployment lays the groundwork for future development of the CIIP structure and provides valuable insight into the operational environment. The deployment need not focus on incident response; it is beneficial to also concentrate on developing local and international relationships, which are vital during the later phases of CIIP structure development.

6.3 Intermediate Development

After the operational environment has been assessed, the development of the national CIIP structure transitions into the intermediate phase. The intermediate phase focuses on the development of the national protection structures, especially the CSIRT and C-SAW teams.

CSIRT Development. The development of the CSIRT is crucial during the intermediate phase. The lessons learned during the initial phase and the small-scale deployment are applied when creating the CSIRT. The principal goals are to strengthen local and international relationships, and define the roles and responsibilities of the CSIRT. The CSIRT should also initiate its operations, and begin to monitor and respond to cyber incidents.

C-SAW Team Deployment. After the CSIRT structure is operational and has the ability to respond to incidents (even in a limited capacity), the C-SAW teams can be integrated into the national structure. This enables small businesses and individuals to gain the benefits of the national CIIP structure.

The C-SAW teams should work with the CSIRT to raise awareness and educate users about threats, vulnerabilities and mitigation strategies. The C-SAW teams should actively grow their constituencies and work within their communities to drive cyber security programs at the grassroots level.

6.4 Mature Development

During this stage of development, the national CIIP structure is fully operational. The CSIRT and C-SAW teams can communicate effectively with their stakeholders and manage incidents. Also, relationships with international peers and industry partners are well established.

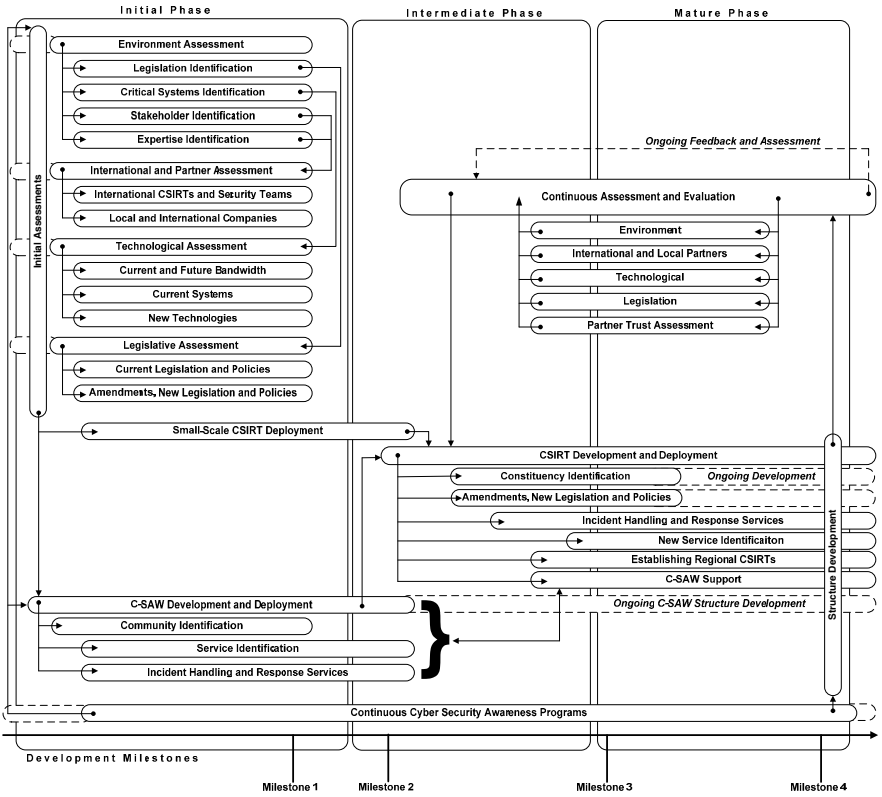


Figure 3. Timeline for deploying a national CIIP structure.

The mature phase does not signify the end of the development cycle. Rather, the national CIIP structure should continuously monitor and adapt to the ever changing environment. Also, CSIRT and C-SAW teams should continue to educate their user bases and ensure that a strong cyber security culture develops over time.

Figure 3 illustrates the CIIP deployment timeline. The timeline spans the four developmental phases, culminating with a mature and robust national CIIP structure.

7. Conclusions

Developing countries have traditionally had poor access to the Internet, and as such have not felt the need to develop national CIIP structures. However, as these countries leverage Internet-based technologies, it is imperative that they develop their own national CIIP structures to ensure reliable operations, incident response and resilience in the face of attacks.

The CIIP framework described in this paper establishes a clear set of phases, goals and outcomes that developing countries can use to establish effective national CIIP structures. Of course, every operational environment is unique. Therefore, it is important that the national CIIP structures accommodate the pertinent features of their environments and continuously evolve with the changing technological and threat landscapes.

Every country must take strong steps to protect its information infrastructure and critical systems. To this end, every national CIIP structure should aim to be all encompassing, open, transparent and publicly available.

References

[1] Akamai Technologies, State of the Internet, vol. 1(4), Cambridge, Massachusetts (`www.akamai.com/stateoftheinternet`), 2008.

[2] Akamai Technologies, State of the Internet, vol. 2(3), Cambridge, Massachusetts (`www.akamai.com/stateoftheinternet`), 2009.

[3] S. Baker, S. Waterman and G. Ivanov, In the Crossfire: Critical Infrastructure in the Age of Cyber War, Technical Report, McAfee, Santa Clara, California, 2010.

[4] N. Brownlee and E. Guttman, RFC2350: Expectations for Computer Security Incident Response, 1998.

[5] I. Ellefsen and S. von Solms, Critical information infrastructure protection in the developing world, in *Critical Infrastructure Protection IV*, T. Moore and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 29–40, 2010.

[6] I. Ellefsen and S. von Solms, C-SAW: Critical information infrastructure protection through simplification, in *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience*, J. Berleur, M. Hercheui and L. Hilty (Eds.), Springer, Boston, Massachusetts, pp. 315–325, 2010.

[7] I. Ellefsen and S. von Solms, The community-oriented computer security, advisory and warning team, *Proceedings of the IST-Africa Conference*, 2010.

[8] European Network and Information Security Agency, Baseline Capabilities for National/Governmental CERTs, Heraklion, Crete, Greece (`www.enisa.europa.eu/act/cert/support/baseline-capabilities`), 2009.

[9] European Network and Information Security Agency, EISAS – European Information Sharing and Alert System for Citizens and SMEs, Heraklion, Crete, Greece (`www.enisa.europa.eu/act/cert/other-work/eisas_folder/eisas_roadmap/at_download/fullReport`), 2011.

[10] L. Harris, CIOs drive expansion into Africa, *ITWeb Brainstorm*, November 11, 2011.

[11] J. Harrison and K. Townsend, An update on WARPs, *ENISA Quarterly Review*, vol. 4(4), pp. 13–15, 2008.

[12] R. Heacock, Internet filtering in Sub-Saharan Africa, Technical Report, OpenNet Initiative, Harvard University, Cambridge, Massachusetts (opennet.net/sites/opennet.net/files/ONI_SSAfrica_2009.pdf), 2009.

[13] International Telecommunication Union, ITU Cybersecurity Work Programme to Assist Developing Countries 2007-2009, Geneva, Switzerland (www.itu.int/ITU-D/cyb/cybersecurity/docs/itu-cybersecurity-work-programme-developing-countries.pdf), 2007.

[14] G. Killcrece, Steps for Creating National CSIRTs, CERT Coordination Center, Carnegie Mellon University, Pittsburgh, Pennsylvania (www.cert.org/archive/pdf/NationalCSIRTs.pdf), 2004.

[15] H. Lipson, Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues, Special Report CMU/SEI-2002-SR-009, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania (www.cert.org/archive/pdf/02sr009.pdf), 2002.

[16] President's Information Technology Advisory Committee, Cyber Security: A Crisis of Prioritization, Report to the President, National Coordination Office for Information Technology Research and Development, Arlington, Virginia (www.nitrd.gov/pitac/reports/20050301_cybersecurity/cybersecurity.pdf), 2005.

[17] S. Song, African undersea cables, Many possibilities, Durbanville, South Africa (manypossibilities.net/african-undersea-cables), 2010.

[18] United States Government Accountability Office, Technology Assessment: Cybersecurity for Critical Infrastructure Protection, GAO-04-321, Washington, DC (www.gao.gov/new.items/d04321.pdf), 2004.

[19] M. West-Brown, D. Stikvoort, K. Kossakowski, G. Killcrece, R. Ruefle and M. Zajicek, Handbook for Computer Security Response Teams (CSIRTs), Handbook CMU/SEI-2003-HB-002, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania (www.cert.org/archive/pdf/csirt-handbook.pdf), 2003.

Chapter 3

INTEGRITY-ORGANIZATION BASED ACCESS CONTROL FOR CRITICAL INFRASTRUCTURE SYSTEMS

Abdeljebar Ameziane El Hassani, Anas Abou El Kalam, and Abdellah Ait Ouahman

Abstract The organization-based access control (OrBAC) model is an access control model that helps evaluate the security policies of organizations. OrBAC affords a high degree of expressiveness and scalability. The model, however, does not readily express integrity constraints. Integrity is one of the most important properties for critical infrastructure systems, mainly due to their criticality and low tolerance of corruption and alterations. This paper describes an extension of OrBAC, called Integrity-OrBAC (I-OrBAC), which models integrity attributes associated with critical infrastructure systems. I-OrBAC facilitates the modeling of multiple integrity levels to express the requirements of different critical infrastructure organizations. An example security policy is presented to demonstrate the expressiveness of the model.

Keywords: Access control, organization-based control, security models, integrity

1. Introduction

The growing sophistication and interconnection of information systems have increased their vulnerability to attacks. This applies especially to critical infrastructures, which are increasingly dependent on information systems but tend not to tolerate disturbances.

Critical infrastructures are assets whose proper functioning is essential to a societal welfare (e.g., energy distribution and transmission, telecommunications and railway infrastructures). These assets often require the collaboration of multiple organizations to receive and/or provide services. In order to protect these assets throughout the various collaborative activities, security policies and enforcement mechanisms are required that clearly identify the needs, vulnerabilities and threats.

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 31–42, 2012.

The security policy of an organization defines guidelines that specify authorized and unauthorized activities. Security models provide mechanisms to evaluate security policies for completeness and adequacy with regard to security properties. Various security models exist for evaluating the confidentiality, integrity and availability of systems. Critical infrastructures, however, have unique characteristics that are not considered in the development of traditional security models. Of special interest are the extensive integrity requirements associated with critical infrastructures.

The organization-based access control (OrBAC) model has been demonstrated to be very effective for specifying security policies of organizations [1]. However, the OrBAC model has certain deficiencies with regard to ensuring integrity. This paper describes an extension of OrBAC, called Integrity-OrBAC (I-OrBAC), which is specifically designed to express integrity requirements in critical infrastructure environments.

2. Background

Security models facilitate the expression and evaluation of security policies. The first security models such as discretionary access control (DAC) [15] and mandatory access control (MAC) [4, 6] enforced a single level of abstraction for representing user permissions. Although they enabled the formal specification of security policies, expressibility was limited and the update functions were complicated and time consuming. Subsequent security models such as role-based access control (RBAC) [11, 12] introduced a second level of abstraction to facilitate manageable update functions and to include dynamic access control rules. Other models support policy specification by integrating notions of obligations [16] and prohibitions [5] to express exceptions.

2.1 OrBAC Security Model

The OrBAC model [1], designed as an extension to RBAC, uses two levels of abstraction to express a security policy: (i) a concrete level; and (ii) an abstract level. The concrete level includes subjects, actions and objects. The abstract level specifies security policies using roles, activities and views. Subjects are abstracted into roles that can perform the same activities (i.e., actions defined by security rules). Objects are similarly abstracted into groups, called views and activities, according to the applicable security rules. Abstract entities enable the expression of organization-specific policies via abstract privileges. Concrete privileges can then be derived to help evaluate the validity of system requests based on situations and conditions. Figure 1 summarizes the various relations and entities in the OrBAC model.

OrBAC adopts a centralized approach; it does not express access control in distributed and collaborative environments. Security models such as PolyOrBAC [2, 3], however, extend OrBAC for access control in collaborative environments. Nevertheless, it is important to further develop OrBAC because it

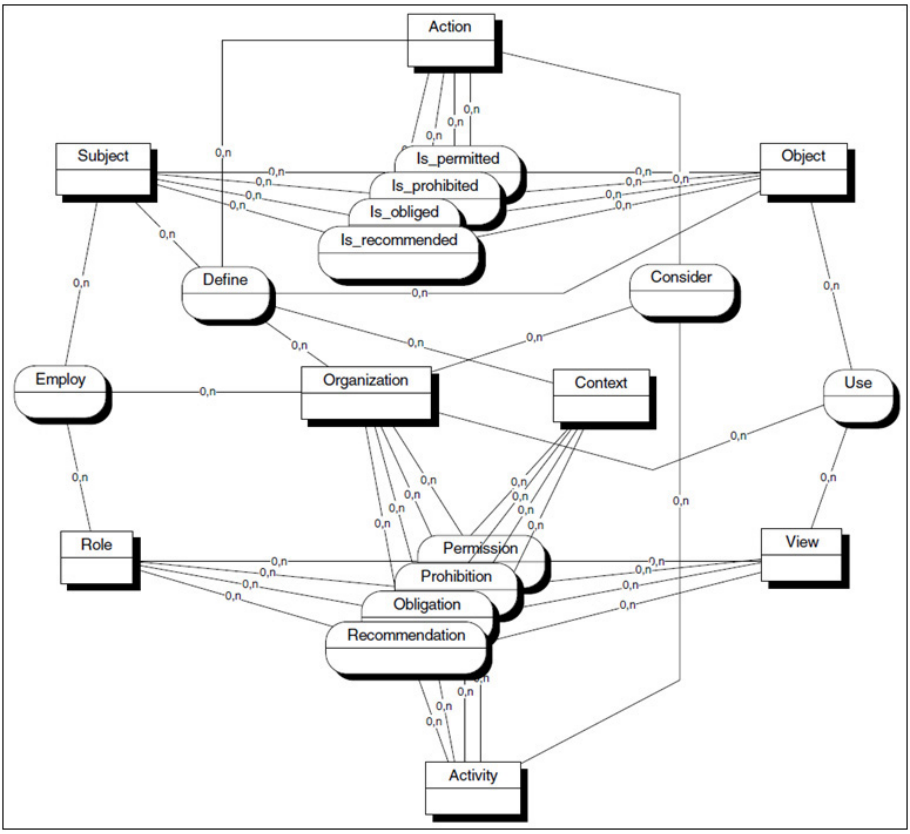


Figure 1. Representation of OrBAC model relations.

expresses organization security policies consistent with the requirements associated with critical infrastructure systems.

Unfortunately, OrBAC does not express policies that enforce data and system integrity. Indeed, subjects inherit privileges granted to roles without *a priori* verification of their empowerment or consideration of their credibility. In addition, all views (i.e., objects) and activities (i.e., actions) are considered to be equally important; this is not representative of operating parameters for critical infrastructure systems. For this purpose, we enrich OrBAC with concepts and mechanisms to help address integrity requirements.

2.2 Integrity Requirements

Critical infrastructures rely extensively on the proper operation and availability of system services. Due to society's reliance on the associated resources, service disruptions can lead to cascading and escalating phenomena with serious financial losses and possibly catastrophic consequences. Although many of

the operating parameters are within the control of asset owners, partnerships, interconnectivity and reliance on assets belonging to other organizations are often necessary. Indeed, the ability to trust data received from other entities is essential to operations. Key aspects associated with data management and trust include secure cooperation, audit and assessment, autonomy and loose coupling, enforcement of permissions, explicit prohibition and obligation rules [2].

This paper focuses on integrity requirements in critical infrastructure environments. Integrity is the property of information to be correct [7]. In this sense, a system must: (i) prevent the unauthorized modification of information (e.g., creation, update and destruction); and (ii) allow the legitimate modification of information. The next section extends the OrBAC model to facilitate the expression of integrity properties for critical infrastructure assets.

3. Integrity-OrBAC Model

According to Krause and Tipton [14], the Biba model [6] was the first security model designed to ensure integrity. Subsequent models (e.g., Goguen-Meseguer [13], Sutherland [14], Clark-Wilson [9], Brewer-Nash (Chinese Wall) [8] and Totel [17]) also provide a means for specifying integrity in security policies. Integrity-specific models, however, are not expressive enough to model the operating parameters, requirements and interactions associated with critical infrastructure assets. OrBAC can model critical infrastructure characteristics, but it does not have the requisite properties for specifying integrity.

Critical infrastructure systems incorporate a wide range of data types that require different integrity requirements depending on the functionality with which they are associated. Additionally, actions within an organization do not all carry the same risks; for example, actions that may involve serious consequences receive higher scrutiny. Moreover, subjects have different expertise and skill levels for performing different tasks. Finally, in addition to technical criteria, subjects should be categorized according to their trustworthiness.

3.1 Assigning Integrity Levels

When developing a critical infrastructure security policy, it is important to properly distinguish several components: (i) information type for each object; (ii) difference between highly sensitive and routine actions; (iii) expertise and skill levels for performing actions; and (iv) degree of trustworthiness associated with each subject. In this sense, the assignment of multilevel integrity values for concrete OrBAC entities must adequately reflect critical infrastructure requirements.

Subject Integrity Levels. The integrity level of each subject is determined on the basis of defined criteria as it relates to the organization. Integrity levels are assigned to the concrete abstraction for each subject. Consider, for example, the role of a "Pilot" in an aviation environment. Not all pilots have

Table 1. Vector representation of subject integrity levels.

	Generalist	Surgeon	Resuscitator	Anesthetist	Therapist	Cardiologist	Neurologist	Trauma	Dentist
Bob	3	3	2	2	–	3	–	–	–
Alice	2	–	–	–	3	–	–	3	–
Eve	3	3	2	2	–	3	–	–	–

the same expertise level; a reputation is earned based on hours of flight time and types of airframe flown. Each pilot is subsequently assigned an integrity level based on the defined parameters. Note that a subject receives a unique integrity level associated with each role performed within the organization. Table 1 presents integrity levels for medical professionals (subjects) using a vector representation.

View and Object Integrity Levels. The integrity level of each object is determined based on the degree of trust inherited from the respective view. The inheritance process affords high expressiveness and also reduces administrative costs. To illustrate this, we revisit the aviation domain. Consider the view *flight_parameters* containing the objects $flight\_plan_x$, $takeoff\_speed_y$ and $altitude_z$; and the view *passengers_data* containing the objects $travel\_class_x$ and $customized\_service_y$. Clearly, the objects contained in the first view are more critical than those in the second view. Thus, the *flight_parameters* view is assigned a higher integrity level than the *passengers_data* view. Note that all the objects in a view inherit the associated integrity level.

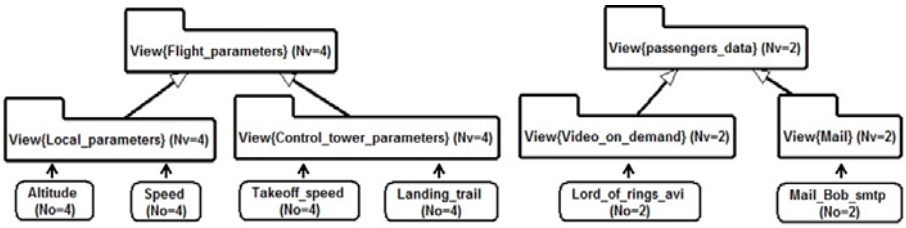


Figure 2. Representation of a view integrity structure.

As illustrated in Figure 2, the views are structured by organization. Consider the object *takeoff_speed*, which is data communicated by airport authorities. The pilot must rely on this information and it is placed in a high integrity level view. Similarly, the *speed* object, as observed from the local parameters of the aircraft, has the same integrity level inherited from *flight_parameters*.

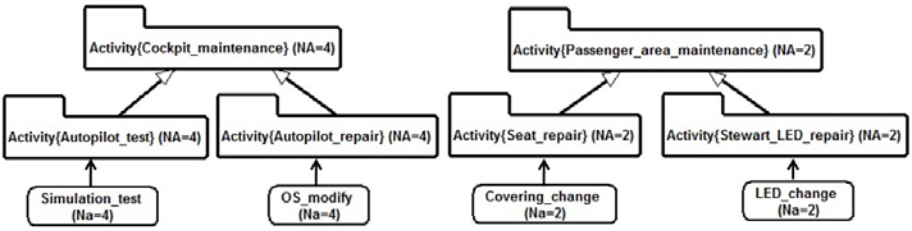


Figure 3. Representation of an activity integrity structure.

Activity and Action Integrity Levels. Actions with the same sensitivity and presenting the same risks to an organization are grouped together in a common activity. As with objects and views, integrity levels are extrapolated to the abstract activity and not to concrete actions. Integrity levels are assigned to activities based on the criticality of the actions they contain and the severity of the impact to the organization. In the aviation example, consider two activities, *cockpit_maintenance* and *passenger_area_maintenance*. Cockpit maintenance requires priority actions to enable safe flight, whereas actions related to the maintenance of passenger area equipment do not have the same criticality. Figure 3 presents an organized representation of activities.

Context Integrity Levels. The ability to understand the context of a system request depends on temporal and spatial characteristics, the purpose and previous actions. Considerations associated with the assignment of context integrity levels include the time the subject requests access, location of the subject, purpose of the access and previous access to the requested object.

Assigning Integrity Levels. In order to quantify integrity levels, meaningful scales must be established that adequately express the associated risks. Figure 4 provides example scales established for roles, views, activities and contexts. Note that other scales can be defined according to the requirements of each critical infrastructure organization.

To ensure overall integrity, privileges are granted by evaluating three parameters: (i) integrity level of the view; (ii) integrity level of the activity; and (iii) integrity level of the context. These three parameters impose constraints on the required integrity level of the subject. An operation is authorized if the subject has the appropriate integrity level.

3.2 I-OrBAC Model Components

I-OrBAC extends OrBAC to incorporate integrity. I-OrBAC is, therefore, based on OrBAC entities, relations, language and axioms. The following expressions summarize the primary OrBAC components used in I-OrBAC. The notation *Org* denotes the organization defined by the security policy. Note that $S \cap O = \emptyset$ in the definitions below.

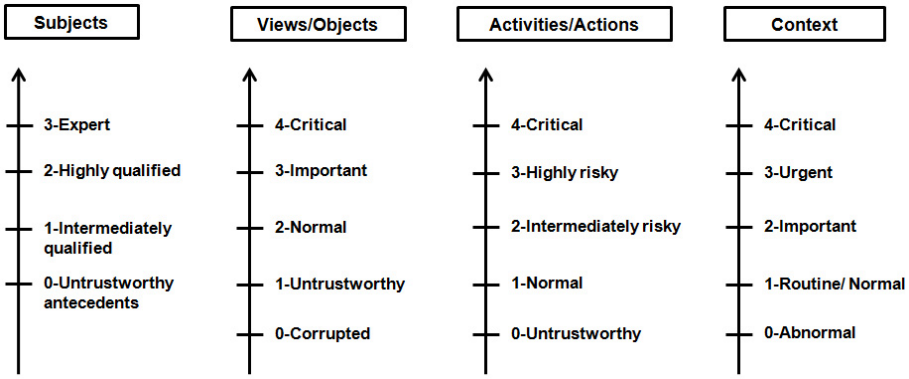


Figure 4. Example integrity level scales.

Specification of entities:

- S: Set of Org subjects
- AC: Set of Org actions
- O: Set of Org objects
- R: Set of Org roles
- AY: Set of Org activities
- V: Set of Org views
- C: Set of contexts related to Org

Relations defining access modes in the abstract level:

- Permission(Org, R, V, AY, C)
- Prohibition(Org, R, V, AY, C)
- Recommendation(Org, R, V, AY, C)
- Obligation(Org, R, V, AY, C)

Relations defining access modes in the concrete level:

- Is_Permitted(S, O, AC)
- Is_Prohibited(S, O, AC)
- Is_Recommended(S, O, AC)
- Is_Obliged(S, O, AC)

Relation affecting a role R to a subject S in Org:

- Empower(Org, S, R)

Relation affecting an object O to a view V in Org:

- Use(Org, O, V)

Relation affecting an action AC to an activity AY in Org:

- Consider(Org, AC, AY)

Relation including the concrete elements of a task – subject S, object O, action AC and context C:

- Define(Org, S, O, AC, C)

Several expressions are introduced to extend OrBAC. These include:

Specification of integrity levels for entities:

- N_S: Set of integrity levels for subjects
- N_V: Set of integrity levels for views
- N_{AY}: Set of integrity levels for activities
- N_C: Set of integrity levels for contexts

Full ordering relation defining "greater than or equal to" for determining the required integrity access level:

- $N_S \geqslant (N_V \times N_V, N_{AY} \times N_{AY}, N_C \times N_C)$

In addition, the relations Empower(), Use(), Consider() and Define() are modified to account for integrity levels.

Value of subject S integrity level in role R:

- Empower(Org, S, R, N_S)

Value of view V integrity level and, by inheritance, object O integrity level:

- Use(Org, O, V, N_V)

Value of activity AY integrity level and, by inheritance, action AC integrity level:

- Consider(Org, AC, AY, N_{AY})

Value of context C integrity level:

- Define(Org, S, O, AC, C, N_C)

4. Access Control Policy Example

This section describes an example from the medical domain. Note, however, that the expressiveness of I-OrBAC enables it to be applied to myriad critical infrastructure assets.

The organization in this example is a hospital *Purpan* that is assessing the treatment of a cancer patient $S = y$ such that y is in the view $V = pat\_surg$, implying the requirement of a surgical intervention. The activity involves critical surgeries $AY = cr\_surg$ and, more specifically, an ablation procedure action $AC = ab\_proc$. The context is considered to be high risk $C = h\_r$ because the patient's life depends on the surgery. Therefore, the constraint that must be evaluated is the integrity level to impose on the doctor $S = x$ who performs the ablation action.

First, the doctor must be a surgeon with the role $R = surg$. It is then necessary to determine the minimum integrity level of the role *surg* that would allow a doctor to perform an ablation. This is accomplished on the basis of integrity levels for *pat_surg*, *cr_surg* and *h_r*.

Consider the following expressions of OrBAC rules corresponding to the *Purpan* security policy:

- Permission(*Purpan*, *surg*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *x*, *surg*)
- Use(*Purpan*, *y*, *pat_surg*)
- Consider(*Purpan*, *ab_proc*, *cr_surg*)
- Define(*Purpan*, *x*, *y*, *ab_proc*, *h_r*)

Permission() enables a surgeon to perform surgery on a patient in the view *pat_surg* associated with context *h_r*. Empower() enables a doctor x to perform in the role *surg*. Use() identifies the patient y in the view *pat_surg*. Consider() includes *ab_proc* as a part of the activity *cr_surg*. Define() provides the context *h_r* of the action *ab_proc*.

The previously established integrity level scales are used to express the rules of the *Purpan* security policy. We consider the context *h_r* as critical with an integrity level N_C=4. The activity *cr_surg* includes the set of critical surgeries and, as a member, the ablation surgery inherits the associated integrity level assigned N_{AY}=4. The view *pat_surg* groups patients who require ablation due to cancer; these patients are at risk of death and require difficult interventions. As such, the patient y is assigned a high integrity level N_V=4.

Given the integrity levels, N_C=4, N_V=4 and N_{AY}=4, we consider that the security policy of *Purpan* only allows surgeons whose integrity level $N_S \geqslant 3$ (i.e., expert subjects in their role) to perform this surgery. The following expressions of I-OrBAC rules articulate these constraints:

- Permission (*Purpan*, *surg*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *x*, *surg*, N_S=3)
- Use(*Purpan*, *y*, *pat_surg*, N_V=4)
- Consider(*Purpan*, *ab_proc*, *cr_surg*, N_{AY}=4)
- Define(*Purpan*, *x*, *y*, *ab_proc*, *h_r*, N_C=4)

A verification of the subject's integrity level N_S is performed to ensure that for x, $N_S \geqslant 3$. Once this is verified, a rule is generated to authorize the action Is_permitted(x, *ab_proc*, y).

4.1 Flexible Integrity Levels

The ability to assign varying integrity levels affords flexibility while protecting objects from unauthorized modification. In the event of an emergency, intervention may be required because of time constraints or the absence of the primary authorized subject. The ability to assign subject integrity levels enables alternative authorizations. For example, assume that the ablation surgery can be performed by a surgeon who specializes in ablation (*s_ab*), an aesthetic surgeon(*s_aesth*) and a general surgeon (*surg*). Individuals in these roles are neither equally skilled nor do they have the same expertise in ablation surgery. It is clear that the most appropriate person to perform the surgery is a surgeon who specializes in ablation, followed by an aesthetic surgeon and, finally, a general surgeon. For this scenario, the security policy imposes different integrity level thresholds for each role in order to perform ablation. The rules are expressed as follows using a six integrity level scale for subjects:

- Permission(*Purpan*, *s_ab*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *bob*, *s_ab*, $N_S = 3$)

or

- Permission(*Purpan*, *s_aesth*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *alice*, *s_aesth*, $n_S = 4$)

or

- Permission(*Purpan*, *surg*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *eve*, *surg*, $N_S = 5$)

The different integrity level thresholds imposed on each role provide a means for enforcing organization guidelines. The security policy strongly recommends that an ablation be performed by a surgeon specialized in ablation. If one is not available, then it is recommended that the ablation be performed by an aesthetic surgeon, followed by a highly skilled surgeon. This flexibility is a variation of the notion of recommendation [10] introduced by OrBAC.

4.2 Integrity Principle Expressed via I-OrBAC

Separation of privilege [7] is a primary security principle that is associated with safeguarding systems and enforcing integrity standards. Separation of privilege states that privileges should be distributed among multiple, independent components such that multiple agreement is necessary to perform an action

(i.e., permission should not be granted based on a single condition). The principle, which is sometimes termed the "two-person rule," ensures high integrity, most notably in highly critical tasks (e.g., launching a nuclear weapon).

Consider a critical context activity that cannot be accomplished without the collaboration of two subjects. Initial constraints require prohibiting all subjects with role R from performing an action AC on their own:

- Prohibition(Org, R, V, AY, C)
- Obligation(Org, $R\hat{}R$, V, AY, C)
- Empower(Org, s_1, R, N_{s1})
- Empower(Org, s_2, R, N_{s2})
- Use(Org, O, V, N_V)
- Consider(Org, AC, AY, N_{AY})
- Define(Org, $s_1\hat{}s_2$, O, AC, C, N_C)
- Is_Obliged($s_1\hat{}s_2$, AC, O)

In the example, two subjects are needed to authorize the action, each of them with the necessary integrity threshold level. As demonstrated, the expressiveness of I-OrBAC enables the articulation of realistic constraints in order to preserve the integrity of objects, actions and contexts.

5. Conclusions

The I-OrBAC extension of the OrBAC model considers integrity aspects and expresses requirements associated with critical infrastructure assets. In particular, I-OrBAC incorporates concepts and components of the OrBAC model while addressing integrity concerns. I-OrBAC quantifies the credibility of subjects along with the criticality of views, activities and contexts in order to preserve the integrity of critical infrastructure assets. The approach supports the modeling of multiple integrity levels to effectively express the requirements of different organizations.

Our future research will focus on secure collaboration in critical infrastructure environments. It will also attempt to develop and evaluate common security policies to determine the most effective implementations for organizations.

References

[1] A. Abou El Kalam, S. Benferhat, A. Miege, R. El Baida, F. Cuppens, C. Saurel, P. Balbiani, Y. Deswarte and G. Trouessin, Organization based access control, *Proceedings of the Fourth International Workshop on Policies for Distributed Systems and Networks*, pp. 120–131, 2003.

[2] A. Abou El Kalam, Y. Deswarte, A. Baina and M. Kaaniche, PolyOrBAC: A security framework for critical infrastructures, *International Journal of Critical Infrastructure Protection*, vol. 2(4), pp. 154–169, 2009.

[3] A. Baina, A. Abou El Kalam, Y. Deswarte and M. Kaaniche, Collaborative access control framework for critical infrastructures, in *Critical Infrastructure Protection II*, M. Papa and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 189–201, 2008.

[4] D. Bell and L. LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation, Technical Report ESD-TR-75-306, MITRE Corporation, Bedford, Massachusetts, 1975.

[5] S. Benferhat, R. El Baida and F. Cuppens, A stratification-based approach for handling conflicts in access control, *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies*, pp. 189–195, 2003.

[6] K. Biba, Integrity Considerations for Secure Computer Systems, Technical Report ESD-TR-76-372, MITRE Corporation, Bedford, Massachusetts, 1977.

[7] M. Bishop, *Computer Security: Art and Science*, Addison-Wesley, Boston, Massachusetts, 2003.

[8] D. Brewer and M. Nash, The Chinese Wall security policy, *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 206–214, 1988.

[9] D. Clark and D. Wilson, A comparison of commercial and military computer security policies, *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 184–195, 1987.

[10] N. Essaouini, A. Abou El Kalam and A. Ait Ouahman, Access control policy: A framework to enforce recommendations, *International Journal of Computer Science and Information Technologies*, vol. 2(5), pp. 2452–2463, 2011.

[11] D. Ferraiolo and D. Kuhn, Role based access control, *Proceedings of the Fifteenth National Computer Security Conference*, pp. 554–563, 1992.

[12] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn and R. Chandramouli, Proposed NIST standard for role-based access control, *ACM Transactions on Information and System Security*, vol. 4(3), pp. 224–274, 2001.

[13] J. Goguen and J. Meseguer, Security policies and security models, *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 11–20, 1982.

[14] M. Krause and H. Tipton, *Handbook of Information Security Management*, Auerbach Publications, Boca Raton, Florida, 1998.

[15] B. Lampson, Protection, *Proceedings of the Fifth Princeton Symposium on Information Sciences and Systems*, pp. 437–443, 1971.

[16] R. Sandhu and J. Park, Usage control: A vision for next generation access control, *Proceedings of the Second International Workshop on Mathematical Methods, Models and Architectures for Computer Network Security*, pp. 17–31, 2003.

[17] E. Totel, J. Blanquart, Y. Deswarte and D. Powell, Supporting multiple levels of criticality, *Proceedings of the Twenty-Eighth IEEE Fault Tolerant Computing Symposium*, pp. 70–79, 1998.

II

CONTROL SYSTEMS SECURITY

Chapter 4

ANALYSIS OF FIELD DEVICES USED IN INDUSTRIAL CONTROL SYSTEMS

John Mulder, Moses Schwartz, Michael Berg, Jonathan Van Houten, Jorge Mario Urrea, and Alex Pease

Abstract A significant portion of the critical infrastructure relies on the proper operation of industrial control system (ICS) field devices. Unfortunately, security solutions for ICS field devices have not progressed sufficiently to address emerging threats. A primary shortfall is the ability to identify device components and analyze their lower level functionality. This paper describes the results obtained from hardware tear-downs of ICS field devices. The results demonstrate the ability to identify key components, analyze device firmware and examine backplane protocols – all necessary steps for the dynamic analysis and development of automated security solutions.

Keywords: Industrial control systems, device analysis, firmware analysis

1. Introduction

Industrial control system (ICS) field devices monitor and control physical processes in the critical infrastructure. Prior to Stuxnet [2], ICS security efforts focused primarily on human-machine interfaces (HMIs) and other supervisory control and data acquisition (SCADA) software. The high-profile attack, however, demonstrates the lack of security associated with ICS field devices. Meanwhile, there has been relatively little research focused on analyzing vulnerabilities associated with these critical assets.

Although many security solutions exist for analyzing software on commodity personal computers, limited tools are available and only a shallow understanding exists of the vulnerabilities related to ICS field devices. The vulnerabilities, however, do exist – initial research on ICS field devices has identified critical security flaws (e.g., hard-coded passwords extracted from firmware images, unauthenticated firmware uploads, multiple unauthenticated interfaces, and weak password hashing) [1, 3–6, 8]. As an example, in January 2012, a coali-

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 45–57, 2012.

tion of security researchers released a set of zero-day vulnerabilities targeting seven embedded ICS devices [7]. Therefore, it is imperative that future security research focus on the analysis of the firmware and hardware implementations.

This paper describes a process for analyzing ICS field devices. Specifically, the components of various programmable logic controllers (PLCs) are examined to derive fundamental attributes that relate to common design characteristics. Note that this analysis is not intended to identify specific vulnerabilities or design flaws. Rather, it seeks a deeper understanding of device design, which is a prerequisite for identifying attack surfaces. The approach is consistent with an adversarial viewpoint – no attempts were made to leverage inside knowledge or obtain vendor cooperation. The analysis of a device was performed by disassembling the device, cataloging its electronic components and creating a functional diagram. The electrical connectivity between the pins was then examined to verify device interconnections. After characterizing the hardware, several techniques were used to extract the firmware, including locating firmware updates, connecting via debug ports and reading the flash memory contents using a chip programmer. Finally, methods were explored for examining the backplanes used for communications between PLC subcomponents.

2. PLC Overview

This paper primarily focuses on PLCs; however, the methodology and results are applicable to a wide range of embedded devices. Indeed, many types of automation equipment have similar control capabilities. For example, the primary automation components used in electrical substations are remote terminal units (RTUs). RTUs perform data aggregation and protocol conversion for the other devices in a substation. Many RTUs, sometimes called real-time automation controllers or substation controllers, execute the same control logic as a PLC. Note that PLC logic is usually written in a language defined in IEC 61131-3 such as Ladder Logic or Instruction List. The primary differences between PLCs and RTUs are in their target markets and configuration details (e.g., input and output specifications).

Modular PLCs, as demonstrated in Figure 1, comprise discrete modules that are connected via a backplane. The processor module reads values from the communications and input/output (I/O) modules, interprets and executes the control logic, and writes values to the communications and I/O modules. The communications modules dissect complex communications-protocol-specific code. Because some control system protocols are extremely complex, communications modules may possess significant processing power and intelligence. I/O modules convert signals between low voltage (3.3V DC or 5V DC), low current (milliamps) control logic and high voltage (24+V DC), high current (amps) process control. Additionally, analog I/O modules contain analog-to-digital converters and digital-to-analog converters.

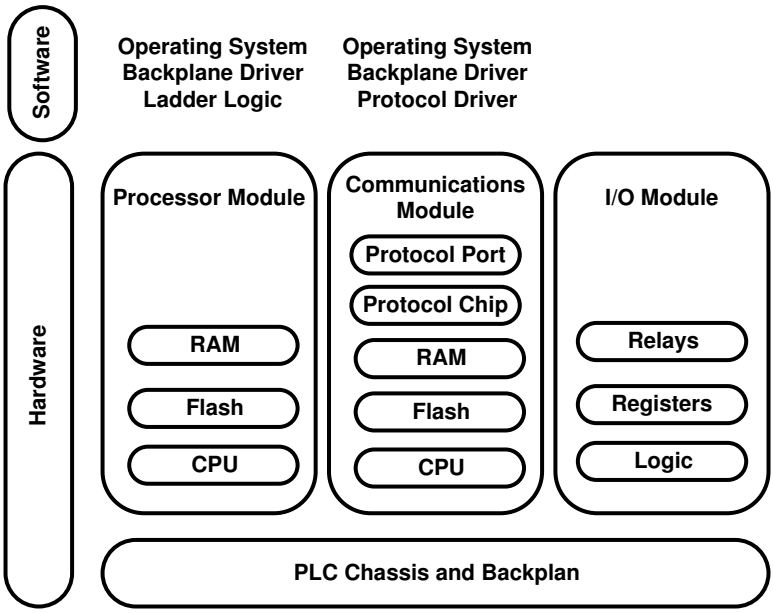


Figure 1. Generic modular PLC.

3. Hardware Analysis

PLC hardware is characterized by disassembling a device, photographing and cataloging the components, researching the parts, and determining the electrical connectivity between components. Although some embedded devices have minimal components that are easy to identify, even a simple PLC has many electronic parts. As such, a heuristic technique was used to discern the components of interest (e.g., memory and processor). The heuristic includes pin-count (higher is more interesting), type of chip package (ball grid array is often used for high-end components), proximity to other interesting parts, and chip markings. Although the heuristic technique approximates the significance of each component, it is adequate for this research effort.

Internet search queries were used for component identification, in particular, to correlate model numbers, brand markings, chip packages and pin counts. After identifying the components, a logical view of the PLC was derived. Component functions and connectivity were discerned using technical data sheets and tracings on the physical circuit card. Findings were verified using an ohmmeter to determine the connections between pins on different chips.

3.1 Example Device Analysis

An Allen-Bradley ControlLogic (Logix) PLC manufactured around 2005 is used to illustrate the methodology (Figure 2). It is one of the most popular

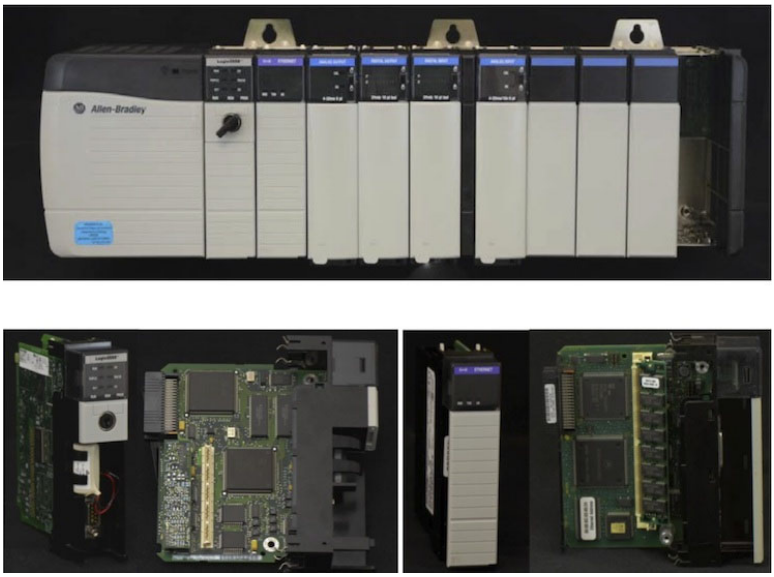


Figure 2. Allen-Bradley Logix PLC.

lines of modular PLCs in North America. The PLC chassis is shown at the top of the Figure 2; the Logix 5555 processor is at the bottom left and the EtherNet/IP modules are at the bottom right. The PLC has a chassis with a power supply and slots for a control module, network communication modules and I/O modules. Figure 3 shows the logical diagram for the connections between the modules and components. This representation clarifies the roles of the various PLC components.

Annotated photographs help clarify the physical layout of the device components – with sufficiently detailed photographs, it is possible to trace the connections on the top and bottom layers of the printed circuit board. Figure 4 shows an annotated photograph of the Logix 5555 processor module and Figure 5 shows an annotated photograph of the EtherNet/IP module. The photographs are each linked to part lists in Tables 1 and 2, respectively. The part name references the data sheet descriptor. The manufacturer part number is a combination of the brand and part number based on the data sheet, when available, or trademarks found on a chip. Note that the markings are the same symbols that are found on the surfaces of the chips.

3.2 General Findings

The majority of field devices we analyzed used production chips from large manufacturers; only a few of the chips were variants produced for a specific vendor. Additionally, we discovered that configurations of flash memory and RAM are consistent with other embedded systems and typically use fairly sim-

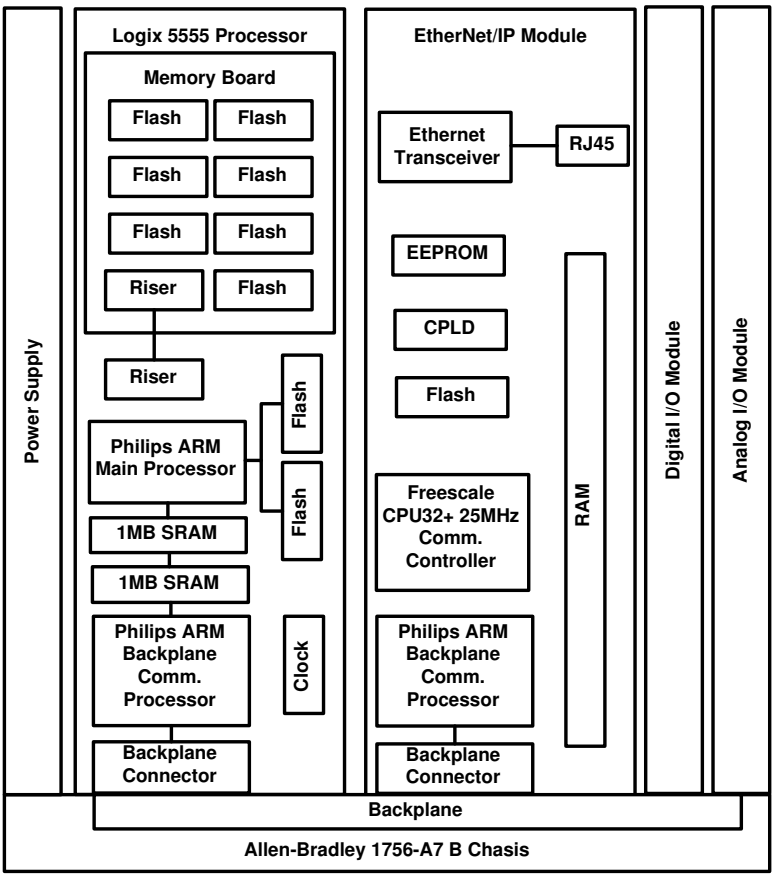


Figure 3. Logical component diagram of the Allen-Bradley Logix PLC.

ple two- to four-layer boards, whereas normal personal computer boards often have seven layers. Despite this simplicity, the interconnections can be difficult to discern primarily because the board layout is driven by efficiency.

The most common processor architectures identified for ICS field devices were ARM, PowerPC and Motorola 68k. However, we also discovered that many devices are based on x86 processors, sometimes using commodity PC/104 form factor embedded computers to provide processing power. Multiple processor architectures on a single board are also common. For example, one device uses a Freescale PowerPC main processor and a separate ARM backplane communication processor. This is not a surprising configuration; however, deep analysis efforts require expertise in multiple architectures.

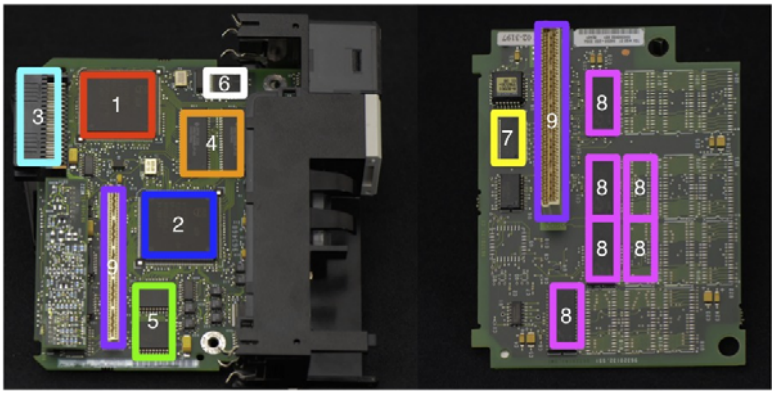


Figure 4. Allen-Bradley Logix 5555 processor module (annotations in Table 1).

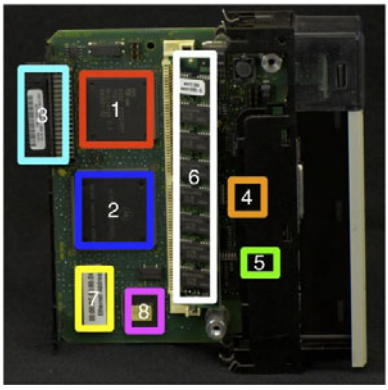


Figure 5. Allen-Bradley EtherNet/IP module (annotations in Table 2).

4. Firmware Analysis

The analysis of PLC device firmware differs from the typical analysis of software (binaries). In general, the toolset for personal computers is not well suited to firmware analysis. Additionally, it may require considerable effort to merely identify the processor and architecture of an embedded device.

The main challenges in understanding embedded devices stem from the diversity of components. Each device uses a different combination of processor architecture, embedded operating system, board design, backplane connector, protocol and logic representation. Many vendors use custom-built components, such as a system-on-a-chip (SOC) main processor, that do not have openly available documentation. The analysis is more difficult because SOCs and custom components are potentially unique, poorly documented and few analysts have experience with them. Unlike "normal" software analysis, the analysis of firmware can depend on chip-specific features. In the case of firmware, vulner-

Table 1. Allen-Bradley Logix 5555 processor module part list (see Figure 4).

	Part Name	**Manufacturer Part Number**	**Markings**
1	Backplane Communications Processor	NXP (Philips Semiconductors) VY21422E2 (Customer-specific product; Discontinued 31 Dec 2005)	PHILIPS ARM VY21422E Y43729Y1 03 KP0250 E MIDRANGE P3E 943631-64
2	Main Processor	NXP (Philips Semiconductors) VY21754A2 (Customer-specific product; Discontinued 31 Dec 2005)	PHILIPS ARM VY21754A Y35737Y1 08 KPr0224 A ARGUS-R2.1 943881-71
3	Backplane Connector		
4	1M High Speed SRAM	Hitachi HM621864HB	JAPAN 0133 HM621864HBLJP-20 00007NN0
5	4 Mb Single Supply Flash Memory	STmicroelectronics M29F040B	M29F040B 45K1 585200210 SINGAPORE
6	Y2K-Compliant Watchdog Real-Time Clock	Dallas Semiconductor DS1501 (May also be branded MAXIM)	DALLAS DS1501YEN 0247A6 045AM
7	32 MB CMOS 5.0V only, Uniform Sector Flash Memory	AMD AM29F032B (Made by Spansion)	AM29F032B -90EI 0113DPB H ©1998 AMD
8	Additional Flash Memory		
9	Riser to Memory Board		

abilities in startup and interrupts are just as interesting as vulnerabilities in network code and application logic.

Table 2. Allen-Bradley Logix 5555 EtherNet/IP module part list (see Figure 5).

	Part Name	Manufacturer Part Number	Markings
1	Backplane Communications Processor	NXP (Philips Semiconductors) VY21086-2 (Customer-specific product; Discontinued 31 Dec 2005)	PHILIPS ARM 0102 y21230Y1 VY21086- Mid_range 2.1 943361-62
2	MC68360 QUad Integrated Communication Controller (QUICC)	Freescale Semiconductor MC68360 (CPU32+ architecture, 25 MHz)	MC68EN360CEM25L OK36E IQAC0049 KOREA
3	Backplane Connector		
4	Enhanced Ethernet Transceiver	Freescale Semiconductor MC68160A	MC68160 AFB HGR0045
5	5V Byte Alterable EEPROM	Xicor X28HC64	XICOR X28HC64JI-90 Cy0047
6	1M x 4-bit CMOS DRAM	Hyundai HY514400A (Eight total)	HY514400A LJ-60 9751C KOREA
7	5V FlashFile Memory	Intel 28F008SA (Sticker covering this chip shows Ethernet MAC address)	PA28F008SA 85 U0200321W (M)(C) '92 '96 Flash
8	High-Density EE CMOS Programmable Logic	Lattice Semiconductor MACH210A	Lattice MACH210A- 10JE -12JI B023PE2

Analysis can proceed quite rapidly when the processor architecture is easy to determine. However, when the processor/operating system combinations are difficult to discern, the initial identification step in device analysis can take a significant amount of effort. To simplify the task, we developed a process for analyzing PLC device firmware that focuses on obtaining firmware images

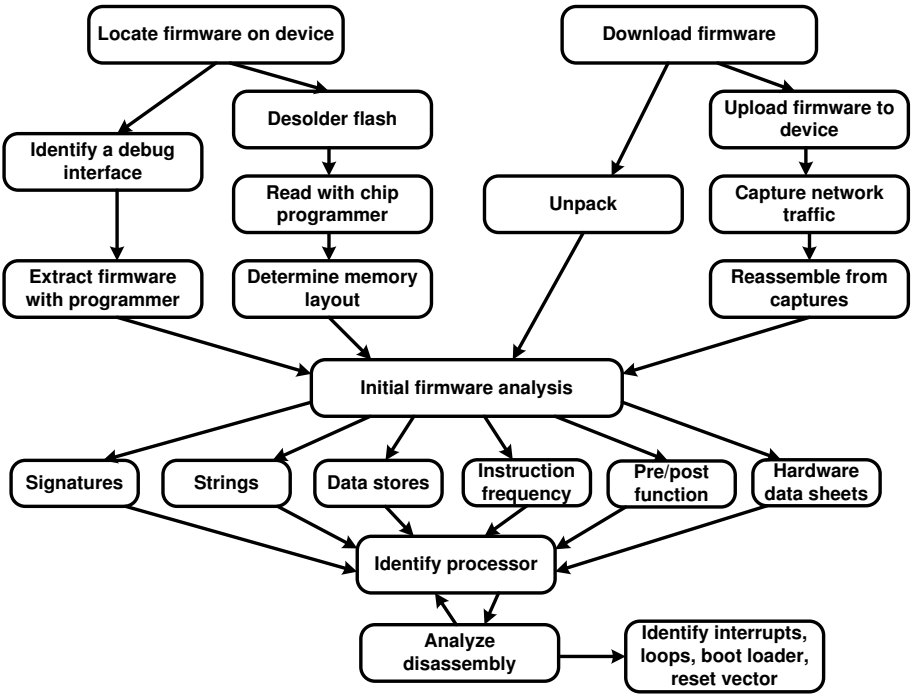


Figure 6. Process for analyzing embedded device firmware.

and identifying the processor architecture. The firmware analysis process is outlined in Figure 6.

4.1 Acquiring Firmware

We investigated four methods for acquiring firmware:

- Read directly from the device using a debug port.
- Read directly from flash memory using a chip programmer.
- Unpack firmware update files.
- Capture network traffic during a firmware update.

Each method for obtaining firmware has inherent difficulties. Connecting via debug ports (e.g., JTAG) works for some devices; however, it is often the case that the interfaces are disabled or non-standard protocols are used. Reading firmware from flash memory consistently worked to obtain the firmware, but unusual memory layouts can make reassembling the entire firmware extremely difficult.

Vendor firmware updates were often not available for the devices of interest to allow the unpacking of update files. In situations where the updates

were available, the firmware contents had to be extracted and reconstructed. Additionally, firmware obtained by capturing network traffic from an update or backup mechanism had to be extracted from the data sections of multiple network packets. Whether they are captured on disk or reconstructed from network captures, the firmware updates obtained from vendor websites often did not represent the actual layout and configuration of the firmware on the devices.

4.2 Identifying the Device Architecture

Identifying the processor and memory architecture of a device was one of the more challenging tasks. In some cases, processors can be identified by their chip markings. However, even when a processor is identified, the memory architectures can vary considerably, especially the flash memory located on the microcontroller, the separate flash and memory boards with a field-programmable gate array (FPGA) interface, and the traceable direct connections between microprocessor and memory. Depending on the design or manufacturing date of a device, it may be possible to guess the likely processors based on their popularity at the time, the relationships between companies, and the processors used in similar devices.

In some cases, we identified the processor type by comparing byte patterns to function calls, register usage and instruction frequency for a likely processor. For example, at the beginning of a function call there is often a store instruction to preserve non-volatile registers and a load instruction is often present at the end of a function. After a processor is tentatively identified, the result can be verified by comparing the firmware code with other code for the processor. The start of function calls, register usage and instruction frequency often facilitate this verification.

5. Communication Backplane Analysis

Analyzing the firmware for every PLC or field device would be extremely time-consuming due to the wide range of hardware and firmware used even within a single product line. However, network protocols are common to a wide range of PLCs. In fact, we discovered that modular PLCs appear to use variants of common network protocols for backplane communications between modules. This means that the backplane presents an avenue for analyzing a wide range of PLCs. Note, however, that backplane analysis is not a replacement for firmware analysis; rather, the two approaches are complementary.

5.1 Identifying Physical Properties

Identifying the pin spacing and layout provides the initial structure of the backplane and allows an analyst to create custom connectors for data collection. A continuity tester and voltmeter can be used to identify various signals on the PLC backplane, which provides insight into the likely locations of ground pins

and some power pins. It is necessary to identify the voltage on each pin to avoid damaging the logic analyzer and to further narrow the pins of interest. Additionally, a voltmeter can be used to identify the operating voltage of each pin on the backplane through startup and normal operation of the PLC.

5.2 Analyzing Pin Logic

After identifying the backplane pins and physical properties, a logic analyzer may be used for deeper analysis. The logic threshold and sampling rate are required to obtain clean captures of normal backplane traffic. These are determined through trial and error. Captures from the logic analyzer may be reviewed to form hypotheses about the use of each pin. Some possible signals of interest include clock-enable, end-of-frame, frame-header, clock and data. The packet timing is first determined by measuring the length of packets and gaps between packets. The backplane traffic is then monitored under several different hardware configurations by removing and reordering modules in the PLC. From these captures, it is possible to identify the modules that created the various packets.

5.3 Translating Backplane Traffic into Bytes

After the data signals have been identified, the analysis of the transmitted data can begin. Logic analyzer software can be used to export the backplane traffic captures to the comma-separated value (CSV) format. The presence of a signal (i.e., voltage above the determined threshold) is represented as a binary one and the absence (i.e., voltage below the threshold) is represented as a binary zero. A simple script can be used to parse the CSV file, translate the binary signals into bytes and identify the header and data sections. If the backplane sends bytes in parallel, it is necessary to identify the order of data pins. This is accomplished by testing different combinations and searching for data bytes that match known patterns (e.g., ASCII, low digits or network protocol headers).

5.4 Analyzing the Backplane Protocol

In order to understand the parser byte output, it is necessary to analyze the backplane protocol. The first step is to determine if the protocol is openly documented. The protocol specification identifies the required fields and unique field values, which can help categorize packets. If software implementations of the protocol are available, they can be used to determine the structure of conversations and packets. Additionally, a comparison between network packet captures and backplane traffic captures can provide insight into the use of some fields; this is especially useful when the captures were taken during the same time period. In our experience, PLC backplanes are often based on network protocols used by PLCs to communicate with external entities.

5.5 Dissecting Backplane Traffic

The final step in backplane analysis is to dissect the backplane traffic. A protocol traffic dissector (e.g., Wireshark) can automatically identify some fields in request and response packets. However, most traffic dissectors do not handle backplane protocols. Therefore, it may be necessary to develop a custom dissector program for analysis.

We have developed a dissector program that consists of a pre-processing script and a protocol field identifier. The pre-processing script reads in binary dumps, identifies packet boundaries, removes collection timestamps and writes ASCII-encoded hex values. The protocol field identifier reads in ASCII-encoded hex values, removes stray line noise, identifies known headers and payloads, and prints a human-readable summary of the packets.

6. Conclusions

Attack techniques that target PLCs will continue to grow in sophistication. As security mechanisms are developed to protect the application layer, attackers will begin to exploit lower levels of abstraction. It is imperative that the security community prepare for this threat and develop automated tools and techniques. The techniques described in this paper can be leveraged to develop automated tools for performing dynamic analyses of PLCs used in the critical infrastructure.

Acknowledgements

This research was supported by the Laboratory Directed Research and Development (LDRD) Program at Sandia National Laboratories, Albuquerque, New Mexico. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy National Nuclear Security Administration under Contract No. DE-AC04-94AL85000.

References

[1] D. Beresford, Exploiting Siemens Simatic S7 PLCs, presented at *Black Hat USA*, 2011.

[2] N. Falliere, L. O'Murchu and E. Chien, W32.Stuxnet Dossier, Symantec, Mountain View, California, 2011.

[3] Industrial Control System Cyber Emergency Response Team (ICS-CERT), Schneider Electric Quantum Ethernet Module Multiple Vulnerabilities, ICS-CERT Alert 11-346-01, Department of Homeland Security, Washington, DC, 2011.

[4] H. Moore, Shiny old VxWorks vulnerabilities, Metasploit (`community.rapid7.com/community/solutions/metasploit/blog/2010/08/p2/shiny-old-vxworks-vulnerabilities`), August 2, 2010.

[5] D. Peck, Security testing, vulnerabilities and exploits in operating systems used in control system field devices, *Proceedings of the SCADA Security Scientific Symposium*, 2010.

[6] D. Peck and D. Peterson, Leveraging Ethernet card vulnerabilities in field devices, *Proceedings of the SCADA Security Scientific Symposium*, 2009.

[7] D. Peterson, Project Basecamp at S4, Digital Bond Blog, Digital Bond, Sunrise, Florida (`www.digitalbond.com/2012/01/19/project-basecamp-at-s4`), January 19, 2012.

[8] R. Santamarta, Reversing industrial firmware for fun and backdoors, Reversemode (`reversemode.com/index.php?option=com_content&task=view&id=80&Itemid=1`), December 12, 2011.

Chapter 5

A FIRMWARE VERIFICATION TOOL FOR PROGRAMMABLE LOGIC CONTROLLERS

Lucille McMinn and Jonathan Butts

Abstract Current supervisory control and data acquisition (SCADA) systems do not have adequately tailored security solutions. Programmable logic controllers (PLCs) in SCADA systems are particularly vulnerable due to a lack of firmware auditing capabilities. Since a PLC is a field device that directly connects to a physical system for monitoring and control, a compromise of its firmware could have devastating consequences. This paper describes a tool developed specifically for verifying PLC firmware in SCADA systems. The tool captures serial data during firmware uploads and verifies it against a known good firmware executable. It can also replay captured data and analyze firmware without the presence of a PLC. The tool does not require any modifications to a SCADA system and can be implemented on a variety of platforms. These features, along with the ability to isolate the tool from production systems and adapt it to various architectures, make the tool attractive for use in diverse SCADA environments.

Keywords: Programmable logic controllers, firmware verification

1. Introduction

The critical infrastructure depends on secure, reliable supervisory control and data acquisition (SCADA) systems that provide critical control, communication and monitoring capabilities over geographically dispersed locations [4, 5]. Because SCADA systems are increasingly interconnected via unsecured networks, security solutions have focused on creating logical and physical boundaries between systems and the network layer [16, 21]. However, even with network isolation, additional attack ingress points have manifested themselves. Indeed, attackers are increasingly leveraging non-traditional means to compromise SCADA systems [18].

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 59–69, 2012.

Current attack response and mitigation tools are inadequately tailored to SCADA systems [26]. In many cases, the available resources are IP network tools that have been adapted to SCADA environments (e.g., packet capture tools, general operating system analysis tools and network-based intrusion detection systems). Although these tools and systems provide protection in a broad sense, tailored security solutions are needed to address emerging threats specific to SCADA systems. Perhaps the most pressing concern is verifying the proper operation of field devices, such as programmable logic controllers (PLCs), that directly monitor and control physical systems. These devices typically operate "below" the network layer and have few security mechanisms. As demonstrated by Stuxnet, manipulation of these devices can have direct physical consequences.

This research describes a tool that helps validate PLC firmware. Firmware, in the most basic sense, is fixed microcode that provides a bridge between the hardware and programmable software on a device. An attacker who can gain access to and manipulate firmware has full control over the functionality of the device and can potentially mask actions from detection. The tool described in this paper helps validate firmware and ensure that any attempt to alter the firmware is detected. The tool is adaptable and portable. Its ability to quickly and safely interface to a computer in order to analyze data sent to a PLC makes it a viable security application in diverse SCADA environments.

2. Background

Most critical infrastructure protection strategies leverage traditional network security constructs. Firewalls and intrusion detection or prevention systems are used to implement a defense-in-depth security strategy. Many SCADA-system-specific solutions engage encryption to achieve confidentiality [9, 17], but such approaches make it more difficult to audit network traffic [19]. Other security solutions often require the modification or addition of system components, which can hinder real-time performance or may be infeasible for large-scale SCADA systems in the field [2, 13, 15, 20].

While network defense is critical to security, network-based attacks are unlikely to be the primary method of exploitation in the future. Stuxnet infiltrated a non-networked environment via a nontraditional vector – a USB device [6]. Indeed, because critical infrastructure assets are high value targets, an advanced persistent threat can be expected to find an input vector to compromise even the most secure system [8, 10, 25]. Considering the variety of non-network-based input vectors, security must be applied beyond the network layer [18].

In SCADA systems, PLCs are field devices that directly connect to physical equipment. The devices control equipment and report data about their operation to remote monitoring stations. The PLCs themselves are typically monitored via a remote human machine interface [3]. As demonstrated by Stuxnet and other recent attacks [7, 23], a PLC under the control of a malicious entity can have devastating effects.

A PLC presents three main targets for attack: hardware, firmware and logic program. Hardware is the lowest layer of abstraction and, at some level, must be trusted. Hardware security requires a trusted supply chain or methods for thoroughly testing the device. In this research, we consider firmware, which includes the PLC operating system, to be the lowest electronically-modifiable layer of a PLC. Note that some PLCs do not have modifiable firmware and others have additional modifiable levels such as a reprogrammable BIOS. Nevertheless, the basic methodology presented in this paper can be used for these architectures, where the lowest modifiable layer is synonymous with firmware.

This research specifically focuses on devices with modifiable firmware and logic program layers. Logic program modifications alter PLC functionality and can be performed fairly easily by accessing the PLC management software. Manipulations of PLC programs, however, can be identified during an inspection. On the other hand, PLC firmware modification is the most intrusive and least detectable attack – there are no easy methods to extract and verify the firmware after it is loaded on a PLC [22]. As shown in Figure 1, the problem is exacerbated by the many potential input vectors that enable firmware alteration (e.g., programming computers, SCADA control systems and firmware update software). Indeed, any access point to a PLC or access to firmware code to be uploaded is an avenue for altering PLC firmware.

3. Tool Design and Evaluation

A tool for verifying that a source device is sending unmodified firmware to a PLC must have three primary features: (i) ability to capture communications data; (ii) ability to analyze captured data; and (iii) ability to determine the validity of the firmware. The tool, which is positioned between the sending device and the PLC, must capture communications data without impacting PLC operations. After the data capture, the tool must analyze the data to determine if the firmware is unmodified.

In the most basic form, the identification of modified firmware is accomplished by comparing the firmware under test with a known good firmware version. For our purposes, we assume a known good baseline version is available for comparison. Note that simple hash comparison for firmware is not as straightforward as checking for modified files in a traditional operating system. Indeed, the requirements to capture and analyze the data as it is being loaded and then perform the comparison render firmware validation nontrivial.

Another important feature is to emulate PLC communications and verify the unmodified firmware independently without the need for a PLC. This type of independent verification is critical to tool portability as it enables implementation on a generic personal computer or mobile computing device for multiple PLC and firmware instances.

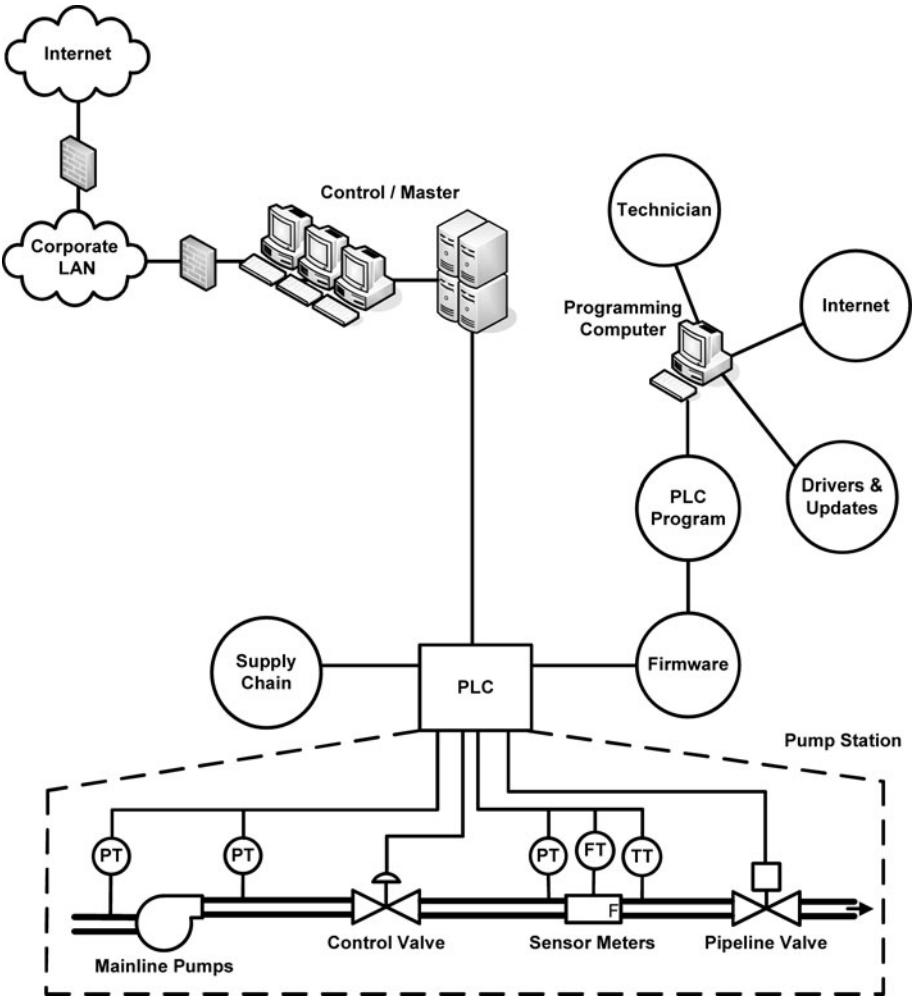


Figure 1. Example non-traditional SCADA system inputs.

3.1 Evaluation Environment

The environment for evaluating the verification tool used a standard Windows XP personal computer and an Allen Bradley FlexLogix 5434 PLC. The personal computer emulated a programming or maintenance system designed to upload the RSLogix firmware. The upload computer had the Rockwell Software RSLogix 5000 suite installed as well as ControlFLASH 9.00.015, the firmware loading program. For the initial baseline capture, the verification tool was connected to the primary communication line via a passive serial adapter tap, enabling the interception of communications data while preserving communications between the uploading computer and PLC. The serial port was configured

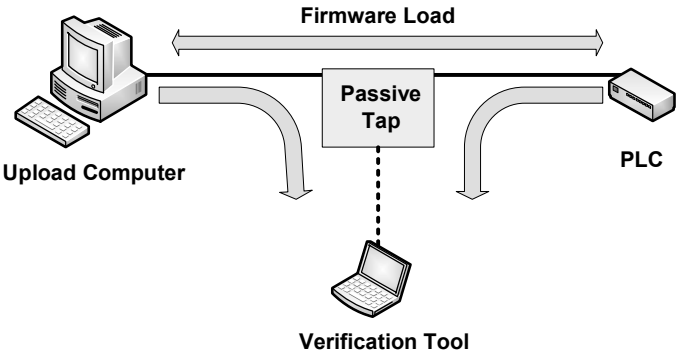


Figure 2. Initial baseline capture configuration.

for the serial data capabilities of the PLC, specifically a baud rate of 19200, eight data bits, no parity and one stop bit. Figure 2 shows the environment used for the initial baseline capture.

During the baseline capture, the uploading computer was connected directly to the PLC with the corresponding DF1 driver controlling communications exchange on the uploading computer side. Firmware version 15.06.01 was loaded on the PLC with the baseline firmware file and the corresponding binary files stored in the ControlFLASH program directory. Each individual ControlFLASH upload was executed in an identical manner using the same version and the same selection method to eliminate variations in the serial data transfer. Data capture started when the ControlFLASH program was opened and ended when the firmware upload was announced as having been completed.

The verification tool was initialized using two captures to create a known good baseline. During this initial capture phase, a firmware load from the uploading computer to the PLC was captured using the serial line. Thus, the verification tool received all the transferred data in a passive manner. The tool used a multithreaded environment to capture serial port data from each line as soon as it became available and stored the data in a binary format. The captured data was then separated into data sent from the firmware uploading computer and reply data sent from the PLC. Note that the data from the uploading computer contained the uploaded firmware bytes.

The entire firmware loading process was executed and captured twice. This allowed the capture program to identify variable protocol packet fields. The two captures were then used to create a baseline for communications. The communications were parsed and the variable bytes were checked against typical protocol field patterns [1, 11, 12, 14]. After all the differences were accounted for, a protocol profile was created, which contained the pattern for communications. The pattern was then applied to the received data in order to emulate future communications.

After the firmware has been loaded on the PLC, the PLC can be taken out of the communications setup and the emulation environment can be implemented

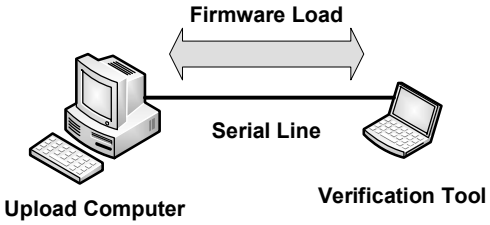


Figure 3. Emulator environment configuration.

by directly connecting the uploading computer to the verification tool (Figure 3). The emulator may then be used to capture and verify subsequent firmware uploads. Note that the emulation environment enables an independent system to evaluate the firmware without the need for the PLC. Indeed, this configuration allows multiple PLC platforms and firmware versions to be validated on a single system without impacting system operations.

3.2 Design

The verification tool was designed in Microsoft Visual Studio to execute on the Windows 7 operating system. The program utilizes two separate serial ports for data capture, where each port is monitored by a thread. The protocol packet field format is not hard coded; instead, it is adapted through analysis of the captured communications data.

The captured data is first grouped into similar packet-like segments. The groupings are based on a start field block consisting of the maximum possible length that occurs a maximal number of times throughout the captured data – both these traits are indicative of a start field block.

A similar method is used to find end field blocks. Note that start/end blocks for the PLC and uploading computer are identified independently. This is because the start/end field blocks may not be the same between the two devices.

After the packet blocks are formed, field mismatches are identified between different communications captures and are accounted for by stateful packet fields. The analysis process checks for typical stateful fields including a BCC error correcting code and a packet identification byte that is sent by the computer and echoed back by the PLC. Escape bytes are also checked because escaping bytes may not be included in the checksum.

The program uses a brute-force parsing method to find optimal field matches. During the evaluation, data captures averaged 2 MB with parsing taking an average of ten minutes (the parsing time would increase for larger data captures). After the initial parsing, the protocol profile that is created is saved and reloaded for future execution to reduce the subsequent run time. Note that the duration of a firmware upload does not change because this is determined by the serial baud rate and the firmware data size.

Given a known good baseline configuration, the emulator is connected to the firmware upload computer directly, replaying PLC responses in accordance with any stateful protocol modifications. The capture program sends responses after each packet is received and can independently induce a firmware upload without the presence of a PLC.

After the uploading is complete, a verification check is performed on the captured communications, validating the received firmware bytes. If all the stateless bytes are the same and the communication matches the baseline communication profile, then it can be concluded that the firmware upload is consistent with the baseline and that the firmware is unmodified.

More thorough checking of the uploaded capture can be performed if the protocol standards are known. All the packets that contain the command to upload firmware packets are parsed to reproduce the loaded firmware file byte-for-byte. This is a more thorough check because it uses prior protocol knowledge to isolate and reproduce the uploaded firmware. The tool has this capability on full-duplex DF1 with the embedded Common Industrial Protocol [1, 11].

4. Evaluation Results

The tool was evaluated using two firmware versions of FlexLogix 5434. The two sets of captured data for protocol analysis were separated by an intermediate period of transferred data to account for any slow changing packet identification variables. All the communications data was successfully captured and parsed into packets using simple optimization algorithms for each variable field. Several versions of modified or incomplete firmware were uploaded, each of which was detected successfully. Additionally, all instances of unmodified firmware were uploaded without any false positive errors.

4.1 Analysis

Parsing, verifying and emulating serial communications between a computer and a PLC require no system modifications, but they provide a thorough security measure. Directly verifying serial data at the last point between the external system and the PLC provides increased assurance that any modifications from the firmware uploading device or any of its input vectors would be identified. Because the firmware uploading procedure follows a deterministic progression, modified firmware can be detected by comparing each new upload against the baseline, even with stateful protocol packet fields. In every test instance, the verification tool was able to identify all the possible outcomes associated with firmware uploads:

- If the firmware is modified to contain at least one more byte or one less byte, then the modified upload contains at least one more byte or one less byte than the baseline.

- If the firmware is modified but still contains the same number of bytes as the previous firmware, then it can be concluded that these bytes are either different from the bytes in the original firmware, or the bytes are in a different order, or both. Therefore, the corresponding packet bytes would be modified in the same manner and must be different from those in the baseline.
- If the firmware is unmodified but the firmware loading program is changed to send modified data, then any changes that correspond to either of the two previous situations are detected.
- If protocol fields are changed so that the load occurs in an identical manner as the baseline but the function codes are changed (e.g., a "store byte" command is changed so that the store is no longer executed, modifying the firmware bytes after they are stored on the PLC instead of prior to upload execution), then the modified function codes are detected as being modified from the baseline function codes.
- If there are no detected differences, then it can be concluded that the uploaded firmware is the same as the original firmware.

4.2 Discussion

The verification tool is currently limited to capturing serial data; however, a similar design can be implemented for other media. While there are many possible configurations for SCADA control devices, the majority of these devices use RS-232, which renders this solution feasible for use on a range of systems [3, 18].

The firmware is the lowest electronically-modifiable level of a SCADA control device. Indeed, firmware validation is the first logical step when considering electronic security. Beyond this, it is also necessary to ensure that the firmware cannot be uploaded remotely by any other means.

PLC memory does not conform with the typical von Neumann architecture. When firmware is loaded, a BIOS writes the uploaded firmware to ROM or flash memory, and the logic program is stored in volatile memory. Without a modified BIOS, the firmware cannot be self-modifying. Executable memory and data are stored separately, preventing a firmware level remote code injection from running PLC firmware.

4.3 Impact

The primary goal of validating firmware extends beyond ensuring that known good firmware is loaded on a PLC – it also helps create a closed system with respect to the PLC. PLCs have the highest level of local control over a SCADA system, so it is critical that they are verified at the basic hardware and software levels before additional security measures are applied at a higher level.

The verification tool offers a novel approach for SCADA security because it requires no system modifications or additions and does not affect the produc-

tion system. Additionally, the verification tool does not introduce new attack vectors to a PLC because the tool is not physically wired to exchange communications with the PLC. Moreover, the tool can operate independently of a PLC.

5. Conclusions

Creating security tools specific to SCADA systems is necessary to maintain and build trust in critical infrastructure systems. The verification tool described in this paper is a viable option for enhancing PLC firmware security. While serial data capture, data verification and emulation are by no means new concepts, the tool combines these concepts in a novel manner that is tailored to SCADA system security. Ideally, a system should be secured from the bottom up. From this point of view, the tool is significant because it helps verify the security of a PLC at the lowest electronically-modifiable level.

Other advantages of the verification tool are that it does not require modifications to the SCADA system, and that it can replay captured data and analyze firmware without the presence of a PLC. These advantages, along with the ability to isolate the tool from production systems and adapt it to various architectures, make the tool attractive for use in diverse SCADA environments.

Note that the views expressed in this article are those of the authors and do not reflect the official policy or position of the U.S. Air Force, Department of Defense or the U.S. Government.

Acknowledgements

This research was partially supported by ICS-CERT under Award HSHQDC-11-X-00089 from the U.S. Department of Homeland Security.

References

[1] Allen-Bradley, DF1 Protocol and Command Set: Reference Manual, Publication No. 1770-6.5.16, Milwaukee, Wisconsin, 1996.

[2] C. Basile, S. Di Carlo and A. Scionti, FPGA-based remote-code integrity verification of programs in distributed embedded systems, *IEEE Transactions on Systems, Man and Cybernetics; Part C: Applications and Reviews*, vol. 42(2), pp. 187–200, 2011.

[3] W. Bolton, *Programmable Logic Controllers*, Elsevier Newnes, Oxford, United Kingdom, 2006.

[4] S. Boyer, *SCADA: Supervisory Control and Data Acquisition*, Instrumentation, Systems and Automation Society, Research Triangle Park, Durham, North Carolina, 2004.

[5] Department of Homeland Security, National Infrastructure Protection Plan, Washington, DC, 2009.

[6] N. Falliere, Exploring Stuxnet's PLC Infection Process, Symantec, Mountain View, California, 2010.

[7] N. Falliere, L. O'Murchu and E. Chien, W32.Stuxnet Dossier, Symantec, Mountain View, California, 2011.

[8] W. Gao, T. Morris, B. Reaves and D. Richey, On SCADA control system command and response injection and intrusion detection, *Proceedings of the eCrime Researchers Summit*, 2010.

[9] G. Gilchrist, Secure authentication for DNP3, *Proceedings of the IEEE Power and Energy Society General Meeting on the Conversion and Delivery of Electrical Energy in the 21st Century*, 2008.

[10] S. Gorman, A. Cole and Y. Dreazen, Computer spies breach fighter-jet project, *Wall Street Journal*, April 21, 2009.

[11] D. Hristu-Varsakelis and W. Levine (Eds.), *Handbook of Networked and Embedded Control Systems*, Birkhauser, Boston, Massachusetts, 2008.

[12] Institute of Electrical and Electronics Engineers, 1815-2010 – IEEE Standard for Electric Power Systems Communications – Distributed Network Protocol (DNP3), Piscataway, New Jersey, 2010.

[13] M. Jakobsson and K. Johansson, Practical and secure software-based attestation, *Proceedings of the Workshop on Lightweight Security and Privacy: Devices, Protocols and Applications*, 2011.

[14] Modicon, Modicon Modbus Protocol Reference Guide, PI-MBUS-300 Revision J, North Andover, Massachusetts, 1996.

[15] T. Morris and K. Pavurapu, A retrofit network transaction data logger and intrusion detection system for transmission and distribution substations, *Proceedings of the IEEE International Conference on Power and Energy*, pp. 958–963, 2010.

[16] National Institute of Standards and Technology, Managing Information Security Risk: Organization, Mission and Information System View, NIST Special Publication 800-39, Gaithersburg, Maryland, 2011.

[17] O. Pal, S. Saiwan, P. Jain, Z. Saquib and D. Patel, Cryptographic key management for SCADA systems: An architectural framework, *Proceedings of the International Conference on Advances in Computing, Control and Telecommunication Technologies*, pp. 169–174, 2009.

[18] M. Schwartz, J. Mulder, J. Trent and W. Atkins, Control System Devices: Architectures and Supply Channels Overview, Sandia Report SAND2010-5183, Sandia National Laboratories, Albuquerque, New Mexico, 2010.

[19] W. Shaw, *Cybersecurity for SCADA Systems*, PennWell, Tulsa, Oklahoma, 2006.

[20] K. Song, D. Seo, H. Park, H. Lee and A. Perrig, OMAP: One-way memory attestation protocol for smart meters, *Proceedings of the Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops*, pp. 111–118, 2011.

[21] K. Stouffer, J. Falco and K. Kent, Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security, NIST Special Publication 800-82, National Institute of Standards and Technology, Gaithersburg, Maryland, 2006.

[22] J. Stradley and D. Karraker, The electronic part supply chain and risks of counterfeit parts in defense applications, *IEEE Transactions on Components and Packaging Technologies*, vol. 29(3), pp. 703–705, 2006.

[23] R. Turk, Cyber Incidents Involving Control Systems, Technical Report INL/EXT-05-00671, Idaho National Laboratory, Idaho Falls, Idaho, 2005.

[24] X. Wang, M. Tehranipoor and J. Plusquellic, Detecting malicious inclusions in secure hardware: Challenges and solutions, *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 15–19, 2008.

[25] G. Wilshusen, Information Security: Cyber Threats and Vulnerabilities Place Federal Systems at Risk, GAO Report GAO-09-661T, Government Accountability Office, Washington, DC, 2009.

[26] G. Wilshusen, Cybersecurity: Continued Attention Needed to Protect Our Nation's Critical Infrastructure, GAO Report GAO-11-865T, Government Accountability Office, Washington, DC, 2011.

Chapter 6

QUANTIFYING CONTROLLER RESILIENCE USING BEHAVIOR CHARACTERIZATION

Henry Bushey, Juan Lopez, and Jonathan Butts

Abstract Supervisory control and data acquisition (SCADA) systems monitor and control major components of the critical infrastructure. Targeted malware such as Stuxnet is an example of a covert cyber attack against a SCADA system that resulted in physical effects. Of particular significance is how Stuxnet exploited the trust relationship between the human machine interface (HMI) and programmable logic controllers (PLCs). Current methods for validating system operating parameters rely on message exchange and network communications protocols, which are generally observed at the HMI. Although sufficient at the macro level, this method does not support the detection of malware that causes physical effects via the covert manipulation of a PLC. This paper introduces an alternative method that leverages the direct analysis of PLC inputs and outputs to derive the true state of SCADA devices. The input-output behavior characteristics are modeled using Petri nets to derive metrics for quantifying the resilience of PLCs against malicious exploits. The method enables the detection of programming changes that affect input-output relationships, the identification of the degree of deviation from a baseline program and the minimization of performance losses due to disruptive events.

Keywords: Programmable logic controllers, behavior characterization, resilience

1. Introduction

Supervisory control and data acquisition (SCADA) systems provide for the automated monitoring and control of major components of the critical infrastructure [1]. SCADA systems were implemented in many industry sectors as early as the 1960s, but security was not a priority at the time. However, recent events – intentional and unintentional – have raised concerns regarding SCADA security [10]. Non-intentional actions have traditionally been addressed using

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 71–83, 2012.

redundant and fault tolerant architectures. Unfortunately, current solutions for dealing with intentional malicious actions are woefully inadequate [12].

A primary risk factor associated with intentional malicious actions is the trend to integrate SCADA systems and enterprise networks to save costs. Indeed, interconnecting critical SCADA systems via LAN and WAN technologies enables numerous attack entry points – from the Internet, internal workstations and communications links between the control center and field sites [10]. As demonstrated by Stuxnet, an attack that propagates from an enterprise network can execute code on field devices such as programmable logic controllers (PLCs) that ultimately results in physical damage to the process system [2].

Current methods for validating the functional parameters of a PLC primarily consider message exchange and network communications protocols, which are generally observed at the human machine interface (HMI). Although this method is sufficient at the macro level, it does not allow for the detection of malware that causes negative physical effects while employing deception techniques to mask the effects from the HMI. Additionally, quantifying the resilience of a PLC requires metrics for assessing its susceptibility to degradation and its ability to recover after an attack. This paper describes a method for detecting programming changes to PLCs by monitoring and characterizing PLC inputs and outputs. Focusing on PLCs at the micro level enables the effects of malicious actions to be observed despite efforts to mask the effects at the HMI, as was the case with Stuxnet.

2. Background

The National Infrastructure Advisory Council [4] defines resilience as the ability to reduce the magnitude and/or duration of disruptive events. An infrastructure is resilient if it can anticipate, absorb, adapt to and/or rapidly recover from a disruptive event. This definition provides a resilience framework with the following four characteristics:

- The ability to monitor the current state to identify deviations from an accepted baseline and anticipate potentially disruptive events.
- The ability to absorb potentially disruptive events by incorporating mechanisms that minimize the amount, if any, of performance loss.
- The ability to adapt by ensuring that contingencies are available that allow for flexible system adjustments to maintain operational availability.
- The ability to recover from disruptive events by incorporating automated or manual mechanisms that allow the system to provide functionality consistent with its baseline.

The method presented in this paper addresses the first two characteristics of the resilience framework and supports mechanisms that help address the last two characteristics. Effectively monitoring and absorbing disruptive events caused by malicious manipulations of PLC functionality requires examination

at the micro level. This level of inspection isolates the PLC and its effects on the external end devices from potentially deceptive reporting to the HMI. The monitoring process is, thus, able to detect the true output states in response to PLC inputs. Analyzing the true output behavior of a PLC provides an accurate method for measuring the absorptive capability of the PLC against disruptive events. Furthermore, the results support the evaluation of contingencies and mechanisms to examine the ability of a system to adapt to and recover from disruptive events.

Other research efforts focus primarily at the macro level and involve static analysis. Queiroz, *et al.* [7] present a model for quantifying SCADA system performance against denial-of-service attacks by determining the probabilistic failures of interdependent nodes of a SCADA system. Germanus, *et al.* [3] present a model in which SCADA system communications use redundant links. Both these methods approach their analysis at a macro level view of the system and are dependent on a trusted HMI for system status.

Shah, *et al.* [9] present a method for verifying PLC executable code. This method utilizes a challenge-response protocol between the verification functions of an untrusted PLC and an external device. While this method approaches the problem at the micro level, the authors have noted that certain logistical issues arise when taking a PLC offline. Note, however, that the method does not protect against timed attacks in which malware is inactive during the verification process. Also, the method does not mitigate the negative effects that occur after malicious code is detected.

Our analysis uses Petri nets derived from PLC inputs and outputs. Petri nets provide a powerful analysis method by abstracting the observed behavior in terms of its graphical and mathematical equivalents. A Petri net C is a four-tuple $C = (T, P, I, O)$ where T is a set of transitions, P is a set of places, I is a set of input functions for each transition and O is a set of output functions for each transition [5]. The Petri nets engaged in our research have been shown to be safe and bounded [6]. This yields finite reachability sets that provide quantifiable data to measure PLC performance during malicious events with respect to the resilience framework.

3. Behavior Characterization Methodology

This section describes the methodology for characterizing the input-output relationships of a PLC. An initial baseline program is established that incorporates PLC programming corresponding to an operational system. After the baseline is established, modifications are made to emulate a PLC infected with malware. Protective schemes are then applied to mitigate the effects of the malware. The enumerated instances of the PLC programs are evaluated to observe deviations in input-output behavior. Petri net models are then utilized to extract metrics that measure PLC security performance with respect to the resilience framework.

3.1 PLC Behavior Characterization

The two primary parameters monitored are: (i) the system input to the PLC; and (ii) the resulting output that translates directly to the physical system end devices (e.g., state of motors, lights and actuators). Because the output behavioral responses are based on the actual status of the physical end devices, the observations can be considered to represent the true state of the system. In order to categorize the observations, PLC interactions with the physical end devices are classified into three input-output response categories:

- **Valid:** Nominal input results in nominal output processes.
- **Degraded:** Nominal input results in deviant but safe output processes. A safe outcome is defined as a non-nominal output response in which system interactions with end devices do not cause catastrophic losses.
- **Unstable:** Nominal input results in deviant and unsafe processes. An unsafe outcome is defined as a non-nominal output response in which system interactions with end devices cause catastrophic losses.

The research environment employed the LogixPro 500 programming software [8] for process emulation and the ProSim II tool [11] for simulation. LogixPro 500 provides a graphical user interface to develop, compile and execute instances of PLC programs with various system operating parameters. The multiple instances demonstrate distinct observable input-output behavior patterns when subjected to malicious attacks. For each instance, four program categories are established:

- **Baseline:** A program that meets the defined process requirements and generates valid input-output responses.
- **Attack Baseline:** A targeted attack applied to the baseline that generates degraded or unstable input-output responses.
- **Protection Baseline:** A protection scheme applied to the baseline that is intended to generate valid input-output responses. The protection scheme produces a fail-safe system state (e.g., flashing red lights at a traffic intersection).
- **Attack Protection Baseline:** A targeted attack applied to the protection baseline that generates valid, degraded or unstable input-output responses.

3.2 PLC Program Development

The various PLC instances establish a basis of observable input-output responses that are modeled and analyzed using Petri nets. The observations obtained from the input-output responses are consistent with black-box analysis; however, using the PLC program in conjunction with the targeted attacks

and protection schemes helps differentiate between the observed behavior and defined nominal process requirements.

A PLC is programmed with a baseline program that is analyzed indirectly by applying baseline input signals and observing the output. The program is analyzed for nominal output behavior (i.e., valid inputs resulting in the nominal output) to demonstrate the PLC baseline interactions with its environment; this establishes the baseline behavior patterns for the model. Next, a series of targeted changes to the baseline program code are implemented that are indicative of alterations by malware. These enumerated versions of the PLC baseline program are also analyzed indirectly by applying the baseline inputs to each enumerated version. The subsequent outputs, which may include inappropriate behavior (e.g., invalid output states), demonstrate the altered interactions of the PLC with its environment; these form the deviated behavior patterns for the model. The purpose of the deviated behavior patterns is to quantify the level of behavioral deviation that is observable in the baseline program. A protection scheme is then applied to the PLC baseline program code. The established protection scheme represents a resilient program that enters a safe state to counter degraded or unstable states. Finally, the targeted malware is applied to the protection scheme in order to evaluate the resilient behavior patterns (i.e., inputs resulting in safe or unsafe outputs). The purpose of the resilient behavior patterns is to quantify the level of behavioral deviation that is observable in the resilient program.

Petri net analysis is performed on the results; this mirrors the states and transitions in the PLC. The absorptive capacity of the protection scheme applied to the PLC is a derivative of the behavioral differences between the baseline and resilient programs. In theory, a fully resilient protection scheme would maximize these measured differences and provide full absorptive capacity against malicious code.

3.3 Petri Net Derivation

This section outlines the methodology for deriving each of the four program categories and the equivalent Petri nets.

1. Establish Baseline Program

(a) Develop a ladder logic program that implements the defined nominal process requirements. The baseline program generates valid system input-output responses.
Consider, for example, a silo plant that fills containers using a conveyer belt and automated sensors. The nominal processes are: bring an empty container into the plant, maneuver the container under the silo valve, fill the container until it is full, and ship the full container from the plant. Figure 1 shows the baseline program for the silo plant.

(b) Abstract the possible combinations of the inputs of the formal ladder logic as transitions T in a Petri net.

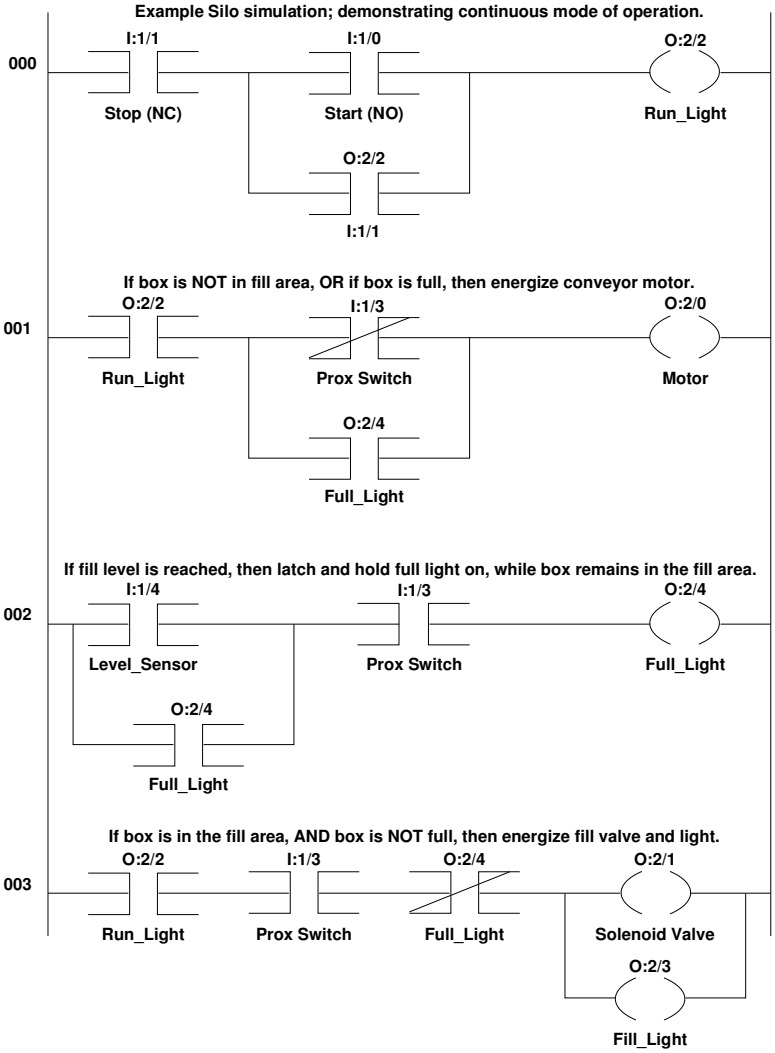


Figure 1. Example baseline ladder logic program for a silo plant.

A transition in a Petri net is defined as a change occurrence in the real-time input of the PLC instance. In this example, four input parameters contribute to the possible transitions in the Petri net: *start, stop, prox switch* and *level sensor* corresponding to I:1/0, I:1/1, I:1/3 and I:1/4, respectively, in Figure 1.

(c) Abstract the possible combinations of the outputs of the formal ladder logic as places P in a Petri net.

A place in a Petri net is defined as a physical state. In the silo plant, five output parameters contribute to the potential places in

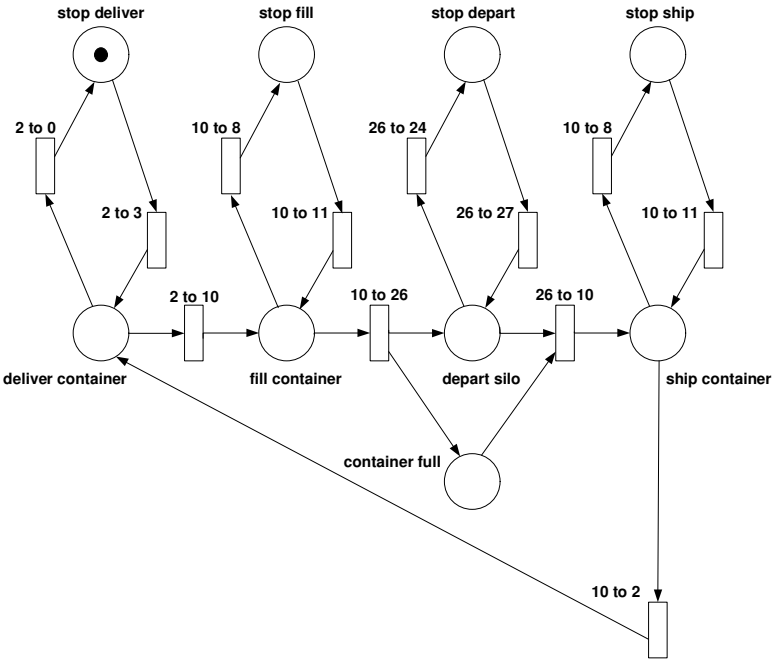


Figure 2. Initial baseline capture configuration.

the Petri net: *motor, solenoid valve, run light, fill light* and *full light* corresponding to O:2/0, O:2/1, O:2/2, O:2/3 and O:2/4, respectively in Figure 1.

(d) Abstract the input and output interdependencies of the formal ladder logic as input and output functions, I and O, for each of the potential transitions in the Petri net.

(e) Combine the subsets derived in Steps 1(a) through 1(d) to define the Petri net $C = (T, P, I, O)$ for analyzing the input-output behavior.

Figure 2 shows the Petri net derived from the example baseline PLC program. The token (black dot) in "stop deliver" denotes that the place is actively manifested as a physical state. From this state, the only transition enabled is Transition 2 to 3. Note that the label "2 to 3" represents the combined decimal input value for the PLC (i.e., Transition 2 to 3 indicates that the system input value changes from decimal value 2 to decimal value 3). Firing Transition 2 to 3 changes the marking of the Petri net and "deliver container" becomes the active physical state. Two transitions are enabled from the active state "deliver container." In cases where there are two or more transitions enabled, the Petri net non-deterministically selects the transition that fires. The resulting Petri net for the baseline

program identifies the possible states that the PLC can reach with valid operations.

2. Establish Attack Baseline

(a) Modify the baseline program ladder logic in a manner consistent with a targeted malicious attack. The attack baseline program generates degraded or unstable system input-output responses.

(b) Repeat Steps 1(a) through 1(e) to produce the equivalent Petri net for the attack baseline PLC program.

An example attack in this scenario targets the proximity sensor so that the silo valve is deceived into remaining open despite a container not being in close proximity to the fill station. The resulting attack manipulates the process to continually dispense product regardless of whether or not a container is in position.

3. Establish Protection Baseline

(a) Modify the baseline program ladder logic to incorporate a protective scheme that mitigates the effects of the attack baseline program. The protection baseline program, which implements fail-safe operation, is intended to generate valid system input-output responses.

(b) Repeat Steps 1(a) through 1(e) to produce the equivalent Petri net for the protection baseline PLC program.

The protection scheme implemented for this example utilizes additional ladder logic to provide a secondary fail-safe check for the silo valve to prevent it from opening when the conveyor belt is in motion.

4. Establish Attack Protection Baseline

(a) Modify the protection baseline program ladder logic to incorporate targeted attacks derived in the attack baseline. The attack protection baseline program generates degraded or unstable input-output responses.

(b) Repeat Steps 1(a) through 1(e) to produce the equivalent Petri net for the attack protection baseline PLC program.

In the case of the baseline attack example, the protection scheme provides fail-safe procedures for safeguarding against the observable effects of the targeted attack that manipulates the proximity sensor. The attack protection baseline program represents the attacks applied to the system with the implemented protection schemes.

In our experiments, we reviewed ten instances and several attacks on the silo process. Each PLC instance generated four programs and the corresponding Petri nets. A total of 40 Petri nets were utilized to mirror the physical states and transitions resulting from the different PLC programs. The quantitative analysis of the 40 Petri nets revealed several consistent findings for assessing PLCs with respect to the resilience framework.

Table 1. Set of tangible states (baseline).

	Container Full	Deliver Container	Depart Silo	Fill Container	Ship Container	Stop Deliver	Stop Depart	Stop Fill	Stop Skip
M0	0	0	0	0	0	1	0	0	0
M1	0	1	0	0	0	0	0	0	0
M2	0	0	0	1	0	0	0	0	0
M3	0	0	0	0	0	0	0	1	0
M4	1	0	1	0	0	0	0	0	0
M5	1	0	0	0	0	0	1	0	0
M6	0	0	0	0	1	0	0	0	0
M7	0	0	0	0	0	0	0	0	1

4. Analysis and Findings

The analysis of the Petri nets yielded several metrics that directly address or indirectly support the four characteristics of the resilience framework. The Petri net simulation application PIPE v4.0 was used to evaluate the Petri nets. The Petri net analysis produced a reachability matrix for each Petri net. These matrices were analyzed to identify comparative metrics that assess true input-output performance with respect to the resilience framework.

4.1 Reachability Matrix

A reachability matrix defines all the possible states in a given Petri net [6]. Table 1 shows the reachability matrix for an example baseline PLC instance. Note that the analysis of the ten instances and the various attacks is consistent with the example used in the discussion. The rows in the table represent the tangible states and the columns represent the places that characterize the states. The elements of each matrix are marked "0" or "1," representing the absence or presence of a token, respectively. For example, State M0 represents the Petri net marking in Figure 2 in which the place "stop deliver" is active. The baseline PLC presented for this instance has eight distinct states.

Table 2. Set of tangible states (attack baseline).

	Container Full	Deliver Container	Depart Silo	Fill Container	Ship Container	Stop Deliver	Stop Depart	Stop Fill	Stop Skip
M0	0	0	0	0	0	1	0	0	0
M1	0	1	0	0	0	0	0	0	0
M2	0	0	0	1	0	0	0	0	0
M3	0	0	0	0	0	0	0	1	0
M4	1	0	1	1	0	0	0	0	0
M5	1	0	0	0	0	0	1	0	0
M6	0	0	0	0	1	0	0	0	0
M7	0	0	0	0	0	0	0	0	1

Table 2 shows an example attack baseline matrix for the PLC instance. Note that State M4 has been altered to incorporate a targeted attack. The attack implemented in this instance causes the silo valve to remain open even though a full container has left the fill station. This state represents a change in the PLC that results in a degraded or unstable operating process, which is not defined in the Petri net corresponding to the baseline program.

Table 3. Comparison of baseline and protection programs.

Attack Instance	Baseline Program	Protection Baseline
1	d/u	fs
2	d/u	fs
3	d/u	fs
4	d/u	fs
5	d/u	fs
6	d/u	fs
7	d/u	fs
8	d/u	d/u
9	d/u	d/u
10	d/u	d/u

4.2 Metrics

The differences between the four program categories form the basis for deriving quantitative metrics. Table 3 summarizes the results of applying the attack instances against the baseline program and the protection baseline program. Note that "d/u" denotes degraded or unstable impact while "fs" denotes system transition to the fail-safe or resilient state.

Attack Instances 1 through 4 alter one state by manipulating one element (e.g., modifying the silo sensor for State M4). Attack Instances 5 through 7 alter one state by manipulating two elements. Attack Instances 8 through 10 alter two states by manipulating two elements for each state. As shown in Table 3, the protection baseline resulted in fail-safe operations for Attack Instances 1 through 7; however, the protection baseline was insufficient for attacks targeting multiple states and multiple elements (Attack Instances 8 through 10).

Comparing these observations with the pairwise differences observed for the ladder logic suggests a correlation between the net changes to the output behavior and the net changes to the ladder logic. As indicated by the results, the most significant metric resulting from the evaluation is the difference observed between the PLC program and the input-output behavior corresponding to the instance. With respect to the resilience framework, this evaluation serves as a self-sufficient metric as well as a complementary metric. Indeed, comparative analysis directly addresses two aspects of resilience (i.e., detecting a change occurrence and quantifying the degree of change occurrence) and supports mechanisms to minimize performance losses due to disruptive events.

The following list summarizes the implications with respect to the resilience framework:

- The ability to anticipate a potentially disruptive event requires the system to be self-aware of its baseline and monitor its current state.

The proposed evaluation identifies when physical input-output relationships deviate from the baseline. This metric may support one of two triggering mechanisms: (i) the number of identified deviations exceeds a threshold; and (ii) a violation against a whitelist of expected outcomes.

- The ability to absorb potentially disruptive events requires the system to have mechanisms in place that minimize the amount, if any, of performance loss.

  The proposed evaluation in combination with the difference in ladder logic changes can help assess the ability of a PLC to absorb disruptive events: (i) if the input-output behavioral difference is zero, then the differences in ladder logic may be used to assess the inherent robustness of the PLC ladder logic program; and (ii) if the input-output behavioral difference is greater than zero, then the differences observed in the input-output behavior may be utilized to assess the overall absorption by the PLC.

- The ability to adapt requires the system to have contingencies that permit flexible system adjustments to maintain operational availability.

  The proposed evaluation supports this feature by providing triggering mechanisms that initiate adaptive processes. Note that the adaptive processes may exist external to the PLC (e.g., requiring coordination with additional hardware and/or software).

- The ability to recover from a disruptive event requires mechanisms, either automated or manual, that enable the system to operate in a manner consistent with its baseline.

  The proposed evaluation supports this feature by providing triggering mechanisms that initiate recovery processes. Note that the recovery processes may exist external to the PLC (e.g., requiring coordination with additional hardware and/or software).

4.3 Extending the Implementation

Although the analysis presented here focuses on an individual PLC program implementation, the methodology can be applied to a more general protection scheme. This is important because the protection mechanism may be an external, and preferably, parallel process. Figure 3 illustrates an example Petri net that models the input-output behavior metric as the primary means for monitoring and detecting state security.

In the example, a monitoring system is incorporated to signal the need to transition to a backup PLC when unstable or degraded operations are detected. The primary PLC executes the baseline program; however, the secondary protective PLC is isolated from direct communications with the SCADA network. If a deviation from the expected behavior is detected, then the secondary PLC triggers a fail-safe operation. During nominal operations, the subnet of the

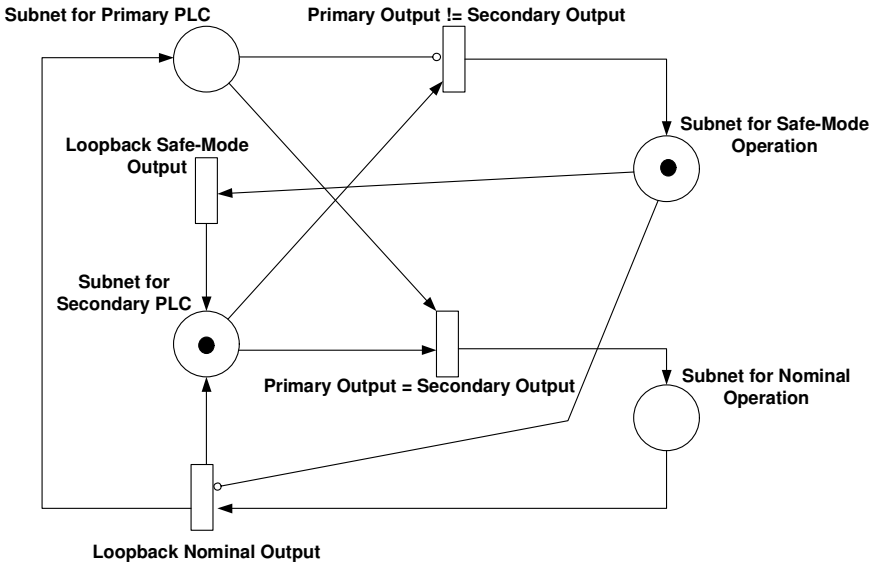


Figure 3. Petri net model for monitoring and detecting security in a fail-safe state.

primary PLC controls process flow. In the case of an input-output behavior deviation, process control is transferred to the subnet of the secondary PLC as represented by the current state in the Petri net. The Petri net illustrates an architecture that supports absorptive, adaptive and recovery features that are triggered primarily by an analysis of the input-output metric.

5. Conclusions

The behavior-based method described in this paper provides a practical means for assessing the security posture of a PLC against malicious code. The resulting framework helps quantify PLC resilience in terms of its ability to monitor, detect and absorb intentional malicious actions. Analyzing the system in real time for nonconforming behavior by the PLC supports attack detection and mitigation. Indeed, metrics from input-output behavior characterization constitute true representations of system state that cannot be deceived by alterations at an HMI or communications channel. Thus, the behavior-based method provides a measure of PLC resilience against malicious code and provides a baseline for quantitatively assessing the security posture. Indeed, analyzing security at the micro level by focusing on field devices and system functions can help prepare for and address future Stuxnet-like attacks.

Note that the views expressed in this article are those of the authors and do not reflect the official policy or position of the U.S. Air Force, Department of Defense or the U.S. Government.

References

[1] Department of Homeland Security, National Infrastructure Protection Plan, Washington, DC, 2009.

[2] N. Falliere, L. O'Murchu and E. Chien, W32.Stuxnet Dossier, Symantec Corporation, Cupertino, California, 2011.

[3] D. Germanus, A. Khelil and N. Suri, Increasing the resilience of critical SCADA systems using peer-to-peer overlays, *Proceedings of the First International Symposium on Architecting Critical Systems*, pp. 161–178, 2010.

[4] National Infrastructure Advisory Council, Critical Infrastructure Resilience Final Report and Recommendations, Department of Homeland Security, Washington, DC, 2009.

[5] J. Peterson, Petri Nets, *ACM Computing Surveys*, vol. 9(3), pp. 223–252, 1977.

[6] J. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice Hall, Upper Saddle River, New Jersey, 1981.

[7] C. Queiroz, A. Mahmood and Z. Tari, Survivable SCADA systems: An analytical framework using performance modeling, *Proceedings of the IEEE Global Communications Conference*, 2010.

[8] Rockwell Automation, RSLogix 500, Milwaukee, Wisconsin.

[9] A. Shah, A. Perrig and B. Sinopoli, Mechanisms to provide integrity in SCADA and PCS devices, *Proceedings of the International Workshop on Cyber-Physical Systems Challenges and Applications*, 2008.

[10] K. Stouffer, J. Falco and K. Kent, Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security, NIST Special Publication 800-82, National Institute of Standards and Technology, Gaithersburg, Maryland, 2006.

[11] The Learning Pit, ProSim II, Whitby, Ontario, Canada.

[12] G. Wilshusen, Cybersecurity: Continued Attention Needed to Protect Our Nation's Critical Infrastructure, GAO Report GAO-11-865T, Government Accountability Office, Washington, DC, 2011.

Chapter 7

USING BLOOM FILTERS TO ENSURE ACCESS CONTROL AND AUTHENTICATION REQUIREMENTS FOR SCADA FIELD DEVICES

Jeffrey Hieb, Jacob Schreiver, and James Graham

Abstract The critical infrastructure cannot operate without SCADA systems; this has made the task of securing SCADA systems a national security priority. While progress has been made in securing control networks, security at the field device level is still lacking. Field devices present unique security challenges and these challenges are compounded by the presence of legacy devices. This paper describes a technique that uses Bloom filters to implement challenge-response authentication and role-based access control in field devices. The approach, which is implemented in an inline security pre-processor, provides for rapid and constant access check times. Experiments involving a prototype device demonstrate that the false positive rate can be kept arbitrarily low and that the real-time performance is acceptable for many SCADA applications.

Keywords: SCADA systems, field devices, security, Bloom filter

1. Introduction

Supervisory control and data acquisition (SCADA) systems and distributed control systems (DCS) are networks of computer systems that provide remote telemetry and control of physical systems and processes. Collectively they are referred to as industrial control systems (ICSs). ICSs play a central role in the operation of many critical infrastructures such as electric power, drinking water, waste water treatment, oil and gas distribution, and industrial manufacturing. A typical ICS comprises a master, one or more field devices and a communications infrastructure. The master or master terminal unit (MTU) processes information received from field devices, presents the information to operators and engineers via a human machine interface (HMI), and sends control directives to field devices. The master and field devices are connected via

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 85–97, 2012.

a communications network, which may include leased serial lines, telephone circuits, cellular networks and UHF/VHF radio. The communication protocols used by the master and field devices are referred to as SCADA protocols.

When these systems were originally deployed, little attention was paid to securing them because the systems were physically isolated and used proprietary hardware, software and communication protocols [2–4, 7, 9]. However, due to network connectivity and convergence, these systems have become highly vulnerable to cyber attacks [7, 9]. Many of the initial efforts to secure these systems use traditional approaches such as firewalls and network intrusion detection systems. While this has been an appropriate first response, Stuxnet and other recent threats underscore the importance of also securing field devices.

We have worked on the task of securing field devices for several years. Our initial effort involved the development of a security-hardened architecture for field devices. Our recent work has focused on developing an in-line security solution for legacy devices using a microkernel based security-hardened architecture. This device, which we call a field device security pre-processor (FD-SPP), provides authentication and role-based access control (RBAC) for legacy field devices. Enforcing RBAC requires checking whether or not a message or operation is allowed, but this involves multiple set membership checks that can have a negative impact on performance.

This paper presents a novel approach using Bloom filters that speeds up field-device-level access control checks to prevent interference with process control operations. In particular, a dual Bloom filter structure is used to minimize security processing while reducing and quantifying the risk of potential attacks. The resulting FD-SPP provides two key security features: authentication using challenge-response authentication and role-based access control enforcement.

2. Background

The field device security pre-processor (FD-SPP) is an in-line device for securing field devices. The FD-SPP uses a security-hardened field device architecture proposed by Hieb, *et al.* [6]. A key advantage of this architecture is that it supports formal verification techniques. The FD-SPP is placed in front of a legacy field device by connecting the communication network interface to the FD-SPP and then connecting the FD-SPP to the field device. A software component running on the MTU/HMI or an external hardware device similar to the FD-SPP works with the FD-SPP to implement security functionality. Figure 1 shows the placement of a FD-SPP in a simple SCADA environment.

To be effective, the FD-SPP needs to implement its security features so that performance is maximized. In addition, the implementation needs to support formal verification techniques. Bloom filters provide a means to achieve both goals. A brief description of the challenge-response authentication scheme is described in Section 2.1 and an overview of the role-based access control technique is presented in Section 2.2. To provide maximum performance, Bloom filters are used to determine if a message is to be challenged and if a received

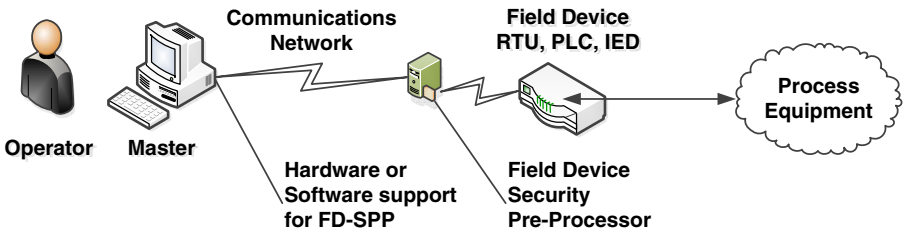


Figure 1. Placement of the FD-SPP in a simple industrial control system.

operation is allowed for the associated user. Section 2.3 provides a brief introduction to Bloom filters.

2.1 FD-SPP Challenge Response Authentication

The incorporation of challenge-response authentication in SCADA protocols is described in detail by Patel [11]. In challenge-response authentication, communicating parties, in this case, an operator or engineer using an MTU/HMI or engineering workstation, and the FD-SPP share a secret (one shared secret for each operator or engineer). When the FD-SPP receives a message from the MTU or workstation, it issues a challenge message in response. The challenge message incorporates a nonce to prevent replay attacks. The MTU/HMI or workstation then builds a challenge-response message, which includes the previous message sent, user identity information, and a hash-based message authentication code (HMAC) that is computed for the message. The HMAC is generated using SHA-256. The HMAC incorporates the shared secret, so only the operator/engineer using the MTU/HMI or workstation can correctly generate the HMAC. The FD-SPP checks the HMAC against the HMAC it calculates. If the two HMACs match, then the message is authenticated and forwarded to the field device.

Due to the nature of ICSs and SCADA protocols, not every message needs to be challenged. For example, reading a coil or analog input has no effect on the field device state, so it is reasonable to have a policy that does not require all messages to be challenged. At runtime, the task of determining whether or not a received SCADA message must be challenged is the responsibility of the access control system discussed in the next section.

2.2 FD-SPP Access Control

In addition to authentication, the FD-SPP provides a simple role-based access control system. In role-based access control (RBAC), users are assigned roles and privileges or capabilities are assigned to the roles. A user may only perform operations assigned to the role possessed by the user. In an ICS setting, there may be different roles for operators, engineers, security administrators and vendors. Grouping privileges or operations by role makes it easier to manage them. For the FD-SPP, each user is assigned a single role, e.g., "op-

erator." When the access control system receives a SCADA message, it must make two decisions: (i) whether or not the message needs to be challenged; and (ii) whether or not the requested operation is to be allowed. Making this decision involves consulting the FD-SPP security policy, which must be developed for each installation by operators, engineers and security administrators. The policy must define roles, assign allowed operations to roles and assign users to roles. If a user attempts to perform an operation that is not associated with the role to which the user is assigned, then the access control system should deny the operation.

2.3 Bloom Filters

A Bloom filter is a probabilistic data structure for determining set membership [1]. Bloom filters have space and time advantages that render them an attractive approach for controlling access to SCADA field devices. The space advantage comes from the fact that a Bloom filter maintains its size no matter how many elements are added to the set. The time advantage arises because there are no loop structures that depend on n (number of elements in the set) to determine if a given element is a member of the set. However, the space and time advantages yield a major disadvantage, false positive errors.

A Bloom filter begins as an empty array of m bits. This empty Bloom filter returns false when any element is checked for membership in the filter. To add an element to the Bloom filter, an element is first passed through k hash functions. The result of each of these hash functions is used to create a position in the Bloom filter array; the bit at each of these positions is set to one. In order to check if an element is in the Bloom filter, the same hash functions are used to create k positions in the Bloom filter. If all the positions in the array have a one, the object is said to be in the Bloom filter. Because of the use of hash functions to add and check entries in the array, collisions in the hash function outputs cause false positive errors. For large values of m, the false positive rate p of a Bloom filter is given by:

$$p = \left(1 - e^{-kn/m}\right)^k. \tag{1}$$

Given n elements in a Bloom filter, the values of k and m can be chosen to achieve any desired false positive rate. In many implementations, m is the nearest power of 2 and k is rounded to the nearest integer m [12].

3. Using a Dual Bloom Filter

In the FD-SPP, Bloom filters are used in two ways: (i) to determine if a requested operation is to be challenged; and (ii) if a requested operation is allowed for the user (assigned to a role) making the request. Initial requests from a source are always challenged by the FD-SPP. After the initial challenge has been met, subsequent operations that are requested are checked to determine if: (i) the user is allowed to request the operation; and (ii) if the operation is

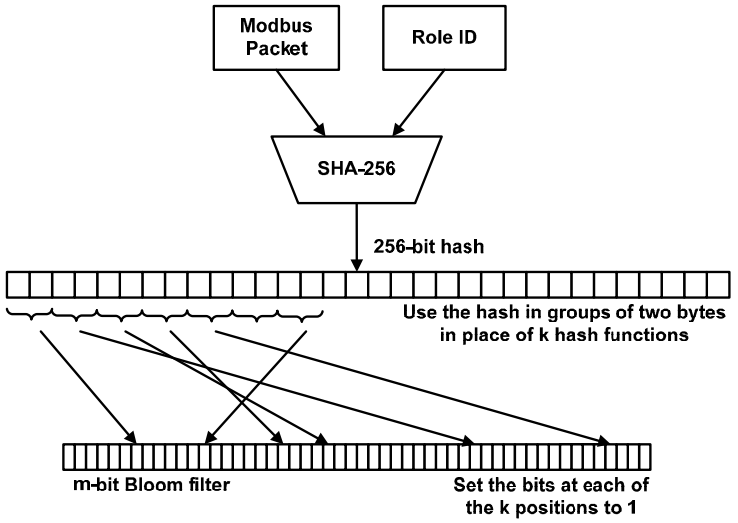


Figure 2. Inserting a Modbus packet in a Bloom filter.

critical and needs to be challenged. The need for quick and efficient processing of SCADA messages by the FD-SPP leads to the possibility of using a Bloom filter to implement the access control checks. The popular Modbus protocol [10] is used for development and testing purposes.

The first step is to create a Bloom filter that implements a given RBAC policy for Modbus operations. We begin by selecting a target false positive rate; in our case, a target false positive rate of 0.01 was used. Next, the bit field size (m) is chosen given the number of entries to be added to the filter. For the given policy, there is an entry for every <role, Modbus Operation> tuple explicit in the policy. For example, if $n = 100$, then $m = 1024$ (nearest power of 2 to 958.5) and $k = 7 \cong 7.0979$.

To add an element to the Bloom filter, the packet and the role ID are combined and hashed using SHA-256. The resulting hash is broken up in order to serve as seven hash functions for the Bloom filter (k=7). The first two bytes of the hash serve as the first hash function, the second two bytes serve as the second hash function, and so on until the seventh hash. Since two bytes are more than necessary to generate a number from 0 to 1023, the lower ten bits of each set of two bytes are used to create the position in the Bloom filter. This scheme has been used by Tripunitara and Carbunar [13]. Using this scheme, role Modbus messages in the policy are added to the Bloom filter one by one as shown in Figure 2.

When a message is received by the FD-SPP, it looks up the user role and hashes the entire message and role to check if the operation is allowed. The sizes and number of hash functions can be varied based on the number of packets and roles desired for the RBAC policy.

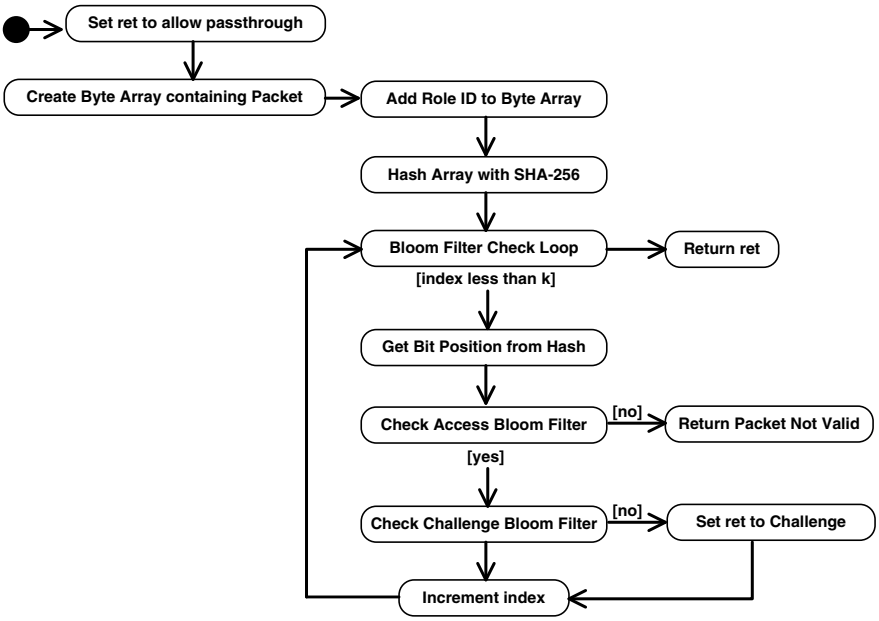


Figure 3. Flow diagram of the access control logic.

A second Bloom filter is used to determine whether or not a message must be challenged. The second Bloom filter can be implemented in two different ways: as a filter containing packets to challenge or as filter containing packets not to challenge. Our rationale for choosing the second approach is provided in Section 4. The second filter has the same number of bits and uses the same hashes as the first filter. However, entries are added to the second filter only if they are not to be challenged. When a requested operation is checked against this Bloom filter, it is hashed in the same way as when adding a new element. The positions are checked in the RBAC Bloom filter first. If all the bits are not equal to one, then the packet is rejected by the RBAC filter. This filter contains all the allowed operations of the field device, which is why it is checked first. If the packet is not rejected, then the same positions are checked in the second Bloom filter. If all the bits in this filter are equal to one, then the packet is allowed to pass on to the field device; if not, a challenge is issued, which authenticates the user requesting the operation. Pass through is allowed in cases where the message is not critical and the sender of the message has recently responded to a challenge. The diagram in Figure 3 presents the procedure for checking entries in the Bloom filter.

The advantage of this approach is that constant time is required for the FD-SPP to check if a given role is allowed to carry out a specific operation and if a message is a critical operation. This is important because it leads to improved processing time for the FD-SPP. However, as described above, a

disadvantage is the false positive rate of Bloom filters. The evaluation in the following section considers performance and the false positive rate.

4. Evaluation

The evaluation was conducted using an FD-SPP prototype for which the Bloom filter approach was implemented on a Gumstix Verdex Pro PXA 270 XScale processor with 64 MB RAM. A simple HMI/MTU was implemented on a separate personal computer using LabView. A virtual serial device was used to place a software component between the MTU/HMI and the serial network interface to provide the complementary FD-SPP support on the MTU/HMI for challenge-response authentication. Modbus messages were collected by sniffing Modbus traffic between the test MTU/HMI and an actual field device before security was added.

Two roles, operator and engineer, were used in the evaluation. Combining the collected Modbus operations with roles resulted in eighteen <role, Modbus Operation> pairs that had to be added to the Bloom filter. Two of the <role, Modbus Operation> message pairs were considered non-critical and were added to the second Bloom filter to indicate that they did not have to be challenged.

4.1 Performance

Bloom filter access checks take a very small amount of time, even when running on minimal hardware. In the case of the Gumstix Verdex Pro XM4 COM, the time required to perform the hash calculation for an element and compare the value with the Bloom filter was about $18\mu s$. On the same device, it took about $15\mu s$ to perform a SHA-256 hash; this time includes only the internal Bloom filter look-up time, not the round trip time from the HMI to the field device. Real-world SCADA installations may require more elements to be added to the Bloom filter, since their policies are likely to be more complicated. However, due to the time-invariant scalability of the Bloom filter, it should be possible to increase the size of the Bloom filter by several orders of magnitude without effecting the access check time. The round trip time was approximately 300ms, which is acceptable in many SCADA applications.

4.2 False Positives

A key issue to be considered when using Bloom filters is the presence of false positives. In this application, a false positive could correspond to a situation where a forged Modbus message inserted by an attacker is accepted unchallenged by the FD-SPP. The false positive rate of the Bloom filter structure indicates the difficulty (or likelihood) that such a message could be found by an attacker. A false positive rate of zero is ideal, but this is not possible with Bloom filters. Instead, the false positive rate must be kept as low as possible while maintaining the speed at which an access check can be performed.

Recall that there are two possibilities in the case of the second Bloom filter: challenged packets are in the Bloom filter or non-challenged packets are in the filter. When the packets are not to be challenged, the false positive rate of the system for non-challenged false positives (successful attacks) is equal to the false positive rate of the second filter. To define this rate, let c represent the number of packets to be challenged, n be the number of packets added to the system, m the length in bits of the filter, k the number of hash functions used, p the false positive rate used to create the first Bloom filter, and r the ratio of packets to be challenged over the total number of packets (c/n). Then, the false positive rate for a Bloom filter containing non-challenged packets is given by:

$$\begin{aligned} p_{non-challenge} &= \left(1 - e^{(\frac{-k(n-c)}{m})}\right)^k = \left(1 - 2^{(\frac{c-n}{n})}\right)^{\frac{ln(2)m}{n}} \\ &= \left(1 - 2^{r-1}\right)^{-\frac{ln(p)}{ln(2)}} \end{aligned}$$

Alternatively, for the Bloom filter containing entries to be challenged, a non-challenged false positive can occur when the packet is accepted by the first filter but rejected by the second. This means that it must have at least one bit that is in the first Bloom filter but not in the second. The maximum odds of this occurring can be calculated as the probability of all but one bit being any of the ones in the first filter multiplied by the ratio of the difference of the number of ones between the filters over the length in bits of the filters. This is given by:

$$\begin{aligned} p_{challenge} &= (1 - e^{\frac{-kn}{m}})^{k-1} * \frac{(m(1 - e^{\frac{-kn}{m}}) - m(1 - e^{\frac{-kc}{m}}))}{m} \\ &= \left(\frac{1}{2}\right)^{\frac{((mln(2))}{(n-1)}} * \left(-\frac{1}{2} + 2^{(\frac{-c}{n})}\right) \\ &= p(-1 + 2^{(\frac{n-c}{n})}) = p(-1 + 2^{1-r}). \end{aligned}$$

This equation shows that the Bloom filter with non-challenged entries has a lower false positive rate because it scales exponentially with the false positive rate of the first Bloom filter while the challenge Bloom filter scales linearly.

After adding 18 entries to the first Bloom filter and two entries to the second Bloom filter (non-challenged operations), it is possible to accurately determine the false positive rate for the specific Bloom filter structure used in this evaluation. Using $n = 18$ in Equation (1), the estimate for the false positive rate is $8.5172 * 10^{-14}$. This is merely the theoretical false positive rate of the approximation of the Bloom filter after 18 entries. The number of ones in the Bloom filter can be used to calculate the actual false positive rate of the Bloom filter. The access control Bloom filter, with $m = 1024$ bits, has 14 bits set to one. Therefore, the probability of any single bit being one is simply 14/1024,

making the actual probability of a false positive in the access control Bloom filter equal to:

$$(\frac{14}{1024})^{7} = 8.9289 * 10^{-14}.$$

The existence of false positives, no matter how small the rate, may appear to indicate that the approach is inappropriate for the FD-SPP access control check. However, we argue that a sufficiently low false positive rate can provide a level of security similar to other techniques. For example, consider the use of symmetric encryption with a key length of n. Brute force attempts to find the key work for sufficiently small n. Similarly, the use of a Bloom filter has a false positive rate relative to k, m and n.

Fortunately, there are several approaches for reducing the false positive rate of the Bloom filter used in this application. The approaches include increasing the size of the Bloom filter and changing the number of hash functions. We assume that the numbers of roles and messages are known before the system is implemented; therefore, the number of bits in the Bloom filter and the number of hash functions can be selected to achieve any desired false positive rate greater than zero. There is also a way to reduce the false positive rate without changing the number of bits or the number of hash functions; this approach is discussed in Section 4.3.

While the false positive rate is central to a security analysis of our approach, there is an important difference between the false positive rate of the Bloom filter structure and the effort required to brute-force a cryptographic secret. If an attacker finds a Modbus message that makes it through the Bloom filter, only this message can be used in an attack (a successful attack would most likely require multiple messages). Additionally, there are only so many Modbus messages that could actually damage a system. For example, only a limited number of messages can write to a particular critical coil. Since the Bloom filter can be checked in advance, analysis can be performed to verify that no messages that can damage the system are non-challenged false positives.

Also, since the attacker would not have access to the Bloom filter, he/she would have to attack the system directly. In the case of the prototype, it takes approximately 200ms to receive a challenge from the security pre-processor. This means that the attacker would have to wait at least 200ms between attempts. If the attacker were to attack the system for an entire day, he/she would be able to perform 432,000 attacks. In Section 4.3, we will show that the actual false positive rate for the implementation can be reduced to $1.65 * 10^{-14}$. Using this final false positive rate, the probability of a successful brute force attack that identifies a single message that could bypass the Bloom filter after one day is just $7.13 * 10^{-9}$.

4.3 Reducing the False Positive Rate

In order to reduce the false positive rate, it is important to first identify the variables that are related to the false positive rate of the Bloom filter. Simply put, if two Bloom filters have the same number of hash functions k and the

same number of bits m, then the only thing that can make the false positive rate any different is the number of ones in the filter. The number of entries in a Bloom filter is based on the number of elements added to the filter multiplied by the number of hash functions minus the number of collisions. Therefore, increasing the number of collisions results in a Bloom filter with a lower false positive rate. Note that this does not imply the use of hash functions that create more collisions; instead, it means using hash functions that collide for the specific values that are added to the Bloom filter. The hash functions must still have uniform results for arbitrary input data, otherwise the bias would yield more false positives, not less. This approach was first described by Hao and colleagues [5].

For example, suppose that two entries A and B are placed in a Bloom filter that uses seven hash functions. Each entry adds seven bits to the Bloom filter, for a total of 14 bits set. Assume that a list of hash functions exists from which the seven hash functions with the most collisions can be selected. In the case of a single collision, 13 bits are added to the Bloom filter instead of 14. Since this value is raised to the power k, the number of collisions can have a large effect on the false positive rate:

$$\frac{13^7}{14^7} = 0.59526.$$

In this case, adding a single collision reduces the false positive rate to nearly 60% of its previous value.

In the more general case, let x be the number of entries in the filter and c the collision percentage that can be invoked. Then,

$$(\frac{x}{m})^k$$

can be reduced to:

$$(\frac{x(1-c)}{m})^k.$$

This means that the false positive rate can be reduced by:

$$1-(1-c)^k.$$

For example, a collision rate of 10% for the known entries of the Bloom filter reduces the false positive rate by more than 50%.

The prototype evaluation system described at the beginning of this section uses seven hash functions, the Bloom filter has 1,024 bits and 14 bits were set to one when the 18 elements were added. This yields the following false positive rate for non-challenged entries:

$$p = (\frac{14}{1024})^7 = 8.9289 * 10^{-14}.$$

Upon searching for collisions between two entries in the second Bloom filter, seven new hash functions with uniform distributions were found. These hash functions had three collisions for the two entries in the challenge-response Bloom filter. This reduces the number of ones from 14 to 11. This yields a collision rate of 0.81514 and more than 80% reduction in the false positive rate using the same number of hash functions and entries, and the same bit length. The new false positive rate for non-challenged false positives is given by:

$$p = (\frac{11}{1024})^7 = 1.65 * 10^{-14}.$$

The new hash functions used are based on the SHA-256 algorithm as in the previous case, so no additional cost for using these hash partitions is imposed.

Since the original Bloom filter was designed for 100 packets whereas only 18 were actually used, the Bloom filter can be further optimized in theory. Using the function above for calculating the number of hash functions shows that 39 hash functions is the optimal number. This number of hash functions produces a false positive rate of $2.4547 * 10^{-44}$. Assuming the worst case situation of no internal collisions, this Bloom filter provides stronger security than a 144-bit cryptographic secret.

Table 1 presents the m and k values that can be used to achieve the targeted false positive rates for different numbers of system messages that would need to be added to an access control Bloom filter structure as described in Section 3. The table entries demonstrate that the approach is viable for SCADA systems with larger numbers of message-role pairs and that acceptable false positive rates can be achieved for these systems.

5. Conclusions

Securing legacy field devices is a challenging task because they have long deployment lifetimes and lack the processing power and memory required to implement security solutions. The field device security pre-processor (FD-SPP), which provides authentication and role-based access control for legacy devices, is a promising in-line security solution for legacy field devices.

The dual Bloom filter approach presented in this paper speeds up access control checks by the FD-SPP to prevent interference with process control operations. In particular, the structure is able to make two access control decisions: (i) whether a message (requested by a user) is allowed or denied; and (ii) whether or not the message is critical and should be challenged. Another advantage of the Bloom filter implementation is that it facilitates the verification of the operation of the entire device. However, a key drawback with the use of a Bloom filter is that it introduces false positive errors. While the errors cannot be eliminated, the analysis indicates that the false positive rate can be made arbitrarily low and that, for sufficiently low false positive rates, the approach is as secure as an n-bit key shared between two parties.

Our future research will focus on the formal verification of access checks and the security code used by the FD-SPP. If formal verification is possible,

Table 1. Parameter values for achieving various false positive rates.

Desired FP Rate	Messages	Percent Challenged	m	k	p
1.00E-13	100	0.50	3,516	24	1.17E-13
1.00E-13	200	0.50	7,033	24	1.16E-13
1.00E-13	300	0.50	10,550	24	1.16E-13
1.00E-13	400	0.50	14,067	24	1.16E-13
1.00E-13	500	0.50	17,584	24	1.16E-13
1.00E-13	100	0.75	2,349	16	1.30E-13
1.00E-13	200	0.75	4,698	16	1.30E-13
1.00E-13	300	0.75	7,047	16	1.30E-13
1.00E-13	400	0.75	9,397	16	1.30E-13
1.00E-13	500	0.75	11,746	16	1.30E-13
1.00E-13	100	0.90	1,597	11	1.14E-13
1.00E-13	200	0.90	3,194	11	1.14E-13
1.00E-13	300	0.90	4,792	11	1.13E-13
1.00E-13	400	0.90	6,389	11	1.13E-13
1.00E-13	500	0.90	7,986	11	1.13E-13
1.00E-20	100	0.50	5,410	37	1.22E-20
1.00E-20	200	0.50	10,821	37	1.22E-20
1.00E-20	300	0.50	16,231	37	1.22E-20
1.00E-20	400	0.50	21,642	37	1.22E-20
1.00E-20	500	0.50	27,052	37	1.22E-20
1.00E-20	100	0.75	3,614	25	1.05E-20
1.00E-20	200	0.75	7,228	25	1.05E-20
1.00E-20	300	0.75	10,842	25	1.05E-20
1.00E-20	400	0.75	14,457	25	1.05E-20
1.00E-20	500	0.75	18,071	25	1.05E-20
1.00E-20	100	0.90	2,457	17	1.06E-20
1.00E-20	200	0.90	4,914	17	1.06E-20
1.00E-20	300	0.90	7,372	17	1.06E-20
1.00E-20	400	0.90	9,829	17	1.06E-20
1.00E-20	500	0.90	12,287	17	1.06E-20

then the resulting quantified false positive rates for FD-SPPs could provide valuable input to risk assessment and risk management efforts for industrial control systems and the critical infrastructure assets in which they are used.

References

[1] B. Bloom, Space/time trade-offs in hash coding with allowable errors, *Communications of the ACM*, vol. 13(7), pp. 422–426, 1970.

[2] T. Brown, Security in SCADA systems: How to handle the growing menace to process automation, *Computing and Control Engineering Journal*, vol. 16(3), pp. 42–47, 2005.

[3] M. Brundle and M. Naedele, Security for process control systems: An overview, *IEEE Security and Privacy*, vol. 6(6), pp. 24–29, 2008.

[4] D. Geer, Security of critical control systems sparks concern, *IEEE Computer*, vol. 39(1), pp. 20–23, 2006.

[5] F. Hao, M. Kodialam and T. Lakshman, Building high accuracy Bloom filters using partitioned hashing, *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 277–288, 2007.

[6] J. Hieb, S. Patel and J. Graham, Security enhancements for distributed control systems, in *Critical Infrastructure Protection*, E. Goetz and S. Shenoi (Eds.), Boston, Massachusetts, pp. 133–146, 2007.

[7] V. Igure, S. Laughter and R. Williams, Security issues in SCADA networks, *Computers and Security*, vol. 25(7), pp. 498–506, 2006.

[8] R. Langner, Stuxnet: Dissecting a cyberwarfare weapon, *IEEE Security and Privacy*, vol. 9(3), pp. 49–51, 2011.

[9] A. Miller, Trends in process control systems security, *IEEE Security and Privacy*, vol. 3(5), pp. 57–60, 2005.

[10] Modbus Organization, Modbus Specification and Implementation Guides, Hopkinton, Massachusetts (`www.modbus.org/specs.php`), 2012.

[11] S. Patel, Secure Internet-Based Communication Protocol for SCADA Networks, Ph.D. Dissertation, Department of Computer Science and Engineering, University of Louisville, Louisville, Kentucky, 2006.

[12] Y. Qiao, T. Li and S. Chen, One memory access Bloom filters and their generalization, *Proceedings of the IEEE International Conference on Computer Communications*, pp. 1745–1753, 2011.

[13] M. Tripunitara and B. Carbunar, Efficient access enforcement in distributed role-based access control (RBAC) deployments, *Proceedings of the Fourteenth ACM Symposium on Access Control Models and Technologies*, pp. 155–164, 2009.

Chapter 8

AGENT INTERACTION AND STATE DETERMINATION IN SCADA SYSTEMS

Thomas McEvoy and Stephen Wolthusen

Abstract Defensive actions in critical infrastructure environments will increasingly require automated agents to manage the complex, dynamic interactions that occur between operators and malicious actors. Characterizing such agent behavior requires the ability to reason about distributed environments where the state of a channel or process depends on the actions of the opposing sides. This paper describes an extension to the Applied π-Calculus for modeling agent behavior in critical infrastructure environments. The utility of the extension is demonstrated via an agent-based attack and defense interaction scenario.

Keywords: Critical infrastructure, agents, state determination, π-Calculus

1. Introduction

Critical infrastructure systems have received significant attention as targets of cyber attacks [4]. Remote attackers, however, face problems in determining the system state due to limitations on communications [5, 7]. At the same time, operators must respond in real-time to sophisticated attacks and may have to make decisions based on partial knowledge of the system state. The outcomes of these interactions may depend on the state (or knowledge of the state) of a single channel or process. We argue that such situations require the deployment of software agents by both sides to automate, in whole or in part, attack and defense strategies.

This paper introduces an extension to the Applied π-Calculus [12] that provides the ability to define and classify agent-based attack and defense strategies. The extension augments a previously-proposed formal adversary capability model [7]. A model of a coordinated attack and defense scenario is presented to illustrate the utility of the extension.

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 99–109, 2012.

2. Background

Software agents often use learning behavior techniques and rely on their perception of the environment to make autonomous decisions [15]. Indeed, in recent years malicious code has exhibited increasingly agent-like behavior, a trend that is likely to continue [10, 11].

Software agents permit an attacker to launch sophisticated attacks, including coordinated attacks on targets. From the offensive perspective, there are several advantages to launching agent-based attacks [2, 3]. Responding to such attacks requires the operator to make dynamic interventions in the face of changing adversary behavior. The difficulty for the operator is exacerbated by the intrinsic nature of critical infrastructure systems, which may require continued operations even in a compromised state [1, 6]. Moreover, the scale and complexity of critical infrastructure systems render it unlikely for a human operator to make the appropriate responses to deal with a coordinated attack.

The ability of software agents to autonomously perform a range of security tasks provides a distinct advantage in countering agent-based attacks [14]. Understanding agent-based attack and defense strategies is key to protecting critical infrastructure systems. However, reasoning about agent-based systems presents a complex set of problems, especially with regard to multiple cooperating agents that "recruit" normally trusted processes to work on their behalf [7]. An extension to the well-known π-Calculus makes it possible to reason about such complex scenarios. The extension also provides a model to examine a distinct class of attacks that are dependent on malicious software agents and not on direct adversary intervention.

3. $G\{\pi\}$-Calculus

The π-Calculus provides a formal mechanism for modeling process actions [7–9, 12]. The associated algebraic theory facilitates formal reasoning, including automating proofs [12]. This section describes the extended goal transform π-Calculus ($G\{\pi\}$-Calculus). Interested readers are referred to [12] for fundamentals of the basic π-Calculus and to [7] for details about the Applied π-Calculus, which is used as the basis of the extension described in this paper.

$||G||_{AgentName}$ is defined by a set of inter-related goals. If G is a goal then:

$$G ::= \mathbf{0}|\pi.G|\nu z\, G|G.G|G + G|G \oplus G|G|G'|!G|[\mathcal{L}]G$$

where the possible actions α of G are defined in Table 1. Note that $\mathcal{L}$ is a first-order logic with equivalence and ordered relations, and π is a capability of the π-Calculus.

Goal actions are defined by the capabilities of the Applied π-Calculus:

$$\pi ::= \bar{x}\langle z\rangle_{\tilde{c}}|x(z)_{\tilde{c}}|\lambda|f(\tilde{v}_{\tilde{c}'}) \supset \tilde{v}'_{\tilde{c}}|[\mathcal{L}]\pi$$

with the semantics defined in Table 2. Goals execute until they invoke another goal, at which point they terminate. G_0 is a reserved label that represents the null goal.

Table 1. $G\{\pi\}$-Calculus syntax.

Term	Semantics
$\mathbf{0}$	Null action
$\pi.G$	Exercise a π-Calculus capability
$\nu z \quad G$	Declare a new goal and its names
$G.G$	Execute goals sequentially
$G + G$	Execute feasible goals in order
$G \oplus G$	Execute exclusive goals
$G \vert G'$	Execute two goals concurrently
$!G$	Replicate goal action
$[\mathcal{L}]G$	Execute a goal based on a first-order logic condition

Table 2. π-Calculus terms.

Term	Semantics
$\bar{x}\langle z\rangle_{\tilde{c}}$	Send a name with characteristics
$x(z)_{\tilde{c}}$	Receive a name with characteristics
λ	A "silent" function
$f(\tilde{v}_{\tilde{c}'}) \supset \tilde{v}'_{\tilde{c}}$	A function over a set of names
$[\mathcal{L}]\pi$	Conditional execution of a capability

The set of system names N comprises channels, constants, message characteristics, variables and function labels. Note that $\tilde{z}$ denotes a vector of names, $\coprod$ denotes a set of concurrent goals or processes, and $\sum$ denotes a sum M over capabilities. A label for an inaction represents a process action that may not be directly observable. For example, if $P := M + \lambda$ is a process, then $\lambda, P \supset 0$ is an inaction or "silent" function of P.

A key characteristic of the model is that processes may be overwritten by messages from another agent. Hence, the outcomes of messages need to be precisely defined. For example, if m is a message and $P := M + \Omega|Q$, then $\Omega, m, P \supset P'$ where P' may be defined arbitrarily. In general, P' behaves like P, except under certain conditions where it executes a different behavior useful to the adversary (i.e., Byzantine behavior).

Destination addresses are characteristics in the example scenario presented in this paper. Assume that $\langle z\rangle_{X_j}$ is used to route the name z to the process X_j. Routing is conditional on the characteristic and, for brevity, is denoted as $\bar{x}_i\langle z\rangle_{[X_j]}$ rather than the more conventional $[z.r = X_j]\bar{x}_{X_j}\langle z\rangle_{X_j}$ where $z.r$ indicates the characteristic routing address of the name z. Hence, any name with the destination address $z.r = X_j$ as a characteristic is routed by $\bar{x}_i$, even

Two agents send malware to selected system nodes.
$||Stage1.SendMalwareToNode_4||_{Launch1} \quad (i = 4)$
$||Stage2.SendMalwareToNode_{12}||_{Launch2} \quad (j = 12)$

The selected nodes are ready to receive the messages.
$||X_4.ReceiveMessage|X_{12}.ReceiveMessage|\coprod_{k=1...15,k\neq i,k\neq j} X_k||_{System}$

After the messages are sent, the launch agents poll for success.
$||Stage1.PollSuccess||_{Launch1}$
$||Stage2.PollSuccess||_{Launch2}$
$||X_4.BecomeMaliciousAgent|X_{12}|\ldots||_{System}$

One of the attacks succeeds and the success is reported to Agent 3.
$||Stage1.ReportSuccessToLaunch3||_{Launch1}]$

The second launch agent continues its initial subversion attempts.
$||Stage2.SendMalware||_{Launch2} \quad (j = 11)$
$||X_4^{'}||_{Agent1}$
$||X_4|X_{11}.ReceiveMessage|\coprod_{k=1...15,k\neq j} X_k||_{System}$

The third agent waits for complete success before it attacks.
$||Stage3.WaitForLaunch1and2 + [Success]LaunchFinalAttack||_{Launch3}$

Figure 1. Initial coordinated attack.

when $\bar{x_i}$ is not the final destination. A proof reduction is indicated using the notation:

$$||Goal.Subgoal.Action||_{Agent} \rightarrow ||Goal.Subgoal.NextAction||_{Agent}$$

where $NextAction$ is any capability or goal invocation. The proof reduction is identical to that of the π-Calculus [12], with the exception that goal labels are used to limit the consideration to the active (i.e., dotted) goals $|| \bullet Goal||$ of each agent.

4. Coordinated Attack

This section models a coordinated attack. Using automated agents, the adversary seeks to manipulate three valves in order to cause a critical failure, while concealing its actions. The first step is to define the goal labels for scenario planning. In fact, without interference in channels and processes, the $G\{\pi\}$-Calculus would be sufficient to prove the outcome of any interactions, provided that the goals are defined precisely.

In the coordinated attack shown in Figure 1, two defined "launch" agents send malware as a name to a system to overwrite various network nodes and transform them into additional malicious agents that work on behalf of the adversary. The scenario has four possible outcomes: (i) success for both agents;

The attack uses two routing processes to forward malware.
$||Stage1.\bar{x}_9\langle g_2\rangle_{X_i}||_{L1} \quad (i = 4)$
$||Stage2.\bar{x}_{10}\langle g_3\rangle_{X_j}||_{L2} \quad (j = 12)$
$||X_9.x_9(z)||_{System}$
$||X_{10}.x_{10}(z)||_{System}$

After the messages are sent, the launch agents wait for success.
$||Stage1.x_{S1}(s)||_{L1}$
$||Stage2.x_{S2}(s)||_{L2}$

The routing agents report either success or failure.
(ϕ is a silent function that indicates failure)
$||X_9.(\bar{x}_i\langle m_2\rangle_{X_i}.\bar{x}_{S1}\langle\top\rangle \oplus \phi.\bar{x}_{S1}\langle\bot\rangle_{S1})||_{Agent0}$
$||X_{10}.((\bar{x}_j\langle m_2\rangle_{X_j}.\bar{x}_{S2}\langle\top\rangle \oplus \phi.\bar{x}_{S2}\langle\bot\rangle_{S2})||_{Agent00}$
$||X_i.x_i(z)||_{System}$
$|X_j.x_j(z)||_{System}$

Case 1: Both attempts succeed.
$||Stage1.x_{S1}(s)||_{L1}$
$||Stage2.x_{S2}(s)||_{L2}$
$||X_9.\bar{x}_{S1}\langle\top\rangle_{S1})||_{Agent0}$
$|X_{10}.\bar{x}_{S2}\langle\top\rangle_{S2}||_{Agent00}$
$||X_i{}'|G||_{Agent2}$
$||X_j{}'|G||_{Agent3}$

Case 2: Either attempt fails and the initial attack continues.

Case 3: Neither attempt succeeds and the attack is aborted.

Figure 2. Reduction of the coordinated attack.

(ii) failure for Agent 1 and success for Agent 2; (iii) success for Agent 1 and failure for Agent 2; and (iv) failure for both agents. Based on the initial outcome, a third launch agent initiates the final part of the attack.

Figure 2 shows the reduction of the coordinated attack. Note that X_i is a system node, g_2 is a message used to infect the system and s is a Boolean variable.

In the example, the launch agents $L1$ and $L2$ send malicious names into the system using channels X_9 and X_{10}, respectively. In turn, these are routed to the target nodes i and j. If the initial subversion succeeds, the newly formed agents flag their success to agent $L3$, which launches the final part of the attack. Note that the messages indicating success or failure may arrive in any order, which may affect the planned outcome.

As demonstrated in Figure 3, the messages update a Boolean predicate a and the final attack launches if the predicate evaluates to $TRUE$. In this instance,

The third launch agent expects a success or failure indicator.
(Counter k keeps track of the number of messages)
$||InitialSuccess.x_{Ad}(u)||_{L3} \quad (k = 0)$

Case 1: Both attacks succeed and counter $k = 2$.
$||InitialSuccess.UpdateAttack.Update(u, a) \supset a'.(k++)\ ||_{L3}$
$\ldots \rightarrow \quad (u = m_2, a = m_2 \wedge \neg m_3, k = 1)$
$||InitialSuccess.UpdateAttack.Update(u, a) \supset a''.(k++)\ ||_{L3}$
$\ldots \rightarrow \quad (u = m_3, a = m_2 \wedge m_3, k = 2)$
$||Stage3||_{L3}$

Case 2: L3 sends a negative flag, but L2 succeeds.
$||InitialSuccess.UpdateAttack.Update(u, a) \supset a'.(k++)\ ||_{L3}$
$\ldots \rightarrow \quad (u = m_2, a = m_2 \wedge \neg m_3, k = 1)$
$||InitialSuccess.UpdateAttack.Update(u, a) \supset a''.(k++)\ ||_{L3}$
$\ldots \rightarrow \quad (u = \neg g_3, a = m_2 \wedge \neg m_3, k = 2)$
$||G_0||_{L3}$

Case 3: L2 and L3 send negative flags.
$||InitialSuccess.UpdateAttack.Update(u, a) \supset a'.(k++)\ ||_{L3}$
$\ldots \rightarrow \quad (u = \neg g_2, a = \neg m_2 \wedge \neg m_3, k = 1)$
$||InitialSuccess.UpdateAttack.Update(u, a) \supset a''.(k++)\ ||_{L3}$
$\ldots \rightarrow \quad (u = \neg g_3, a = \neg m_2 \wedge \neg m_3, k = 2)$
$||G_0||_{L3}$

Case 4: L3 succeeds, but L2 fails.
$||InitialSuccess.UpdateAttack.Update(u, a) \supset a'.(k++)\ ||_{L3}$
$\ldots \rightarrow \quad (u = \neg g_2, a = \neg m_2 \wedge \neg m_3, k = 1)$
$||InitialSuccess.UpdateAttack.Update(u, a) \supset a''.(k++)\ ||_{L3}$
$\ldots \rightarrow \quad (u = m_3, a = \neg m_2 \wedge m_3, k = 2)$
$||G_0||_{L3}$

Figure 3. Determining if the final attack should be executed.

the update order is not relevant to the outcome. The result of Case 1 is that the final part of the attack is launched by L3. At this point, L3 sets the target valve to *Steady* and signals the other two agents to set their target valves to *Open* and *Closed*. The attack is concluded by masking the signal from the operators as demonstrated in Figure 4; this serves to conceal the attack.

5. Distributed Detection

For the defense strategy, an operator employs observer agents for state determination and trusted routes for alerts regarding critical conditions. As described in [8, 9, 13], each network node that receives a message adds its address

The first agent sets a valve to a steady value.
$||X_k{}'|SetSteady.set(u, Steady) \supset u'||_{Agent1}$
$||X_i|Waitfor1||_{Agent2}$
$||X_j|Waitfor2||_{Agent3}$

$||X_k{}'|SetSteady.Send.\bar{x}_s\langle u'\rangle_{C2}||_{Agent1}$
$||X_i|Waitfor1||_{Agent2}$
$||X_j|Waitfor2||_{Agent3}$

The instruction is sent to the controller.
$||X_k{}'||FlagSteady||_{Agent1}$
$||X_i|Waitfor1||_{Agent2}$
$||X_j|Waitfor2||_{Agent3}$

Messages are received.
$||X_k{}'|FlagSteady.x_s(u)||_{Agent1}$
$||X_i|Waitfor1||_{Agent2}$
$||X_j|Waitfor2||_{Agent3}$

After the flag is steady, the next agent is signaled.
$||X_k{}'|FlagSteady.[u = Steady]\bar{x}_s\langle s\rangle_{X_5}.G_0||_{Agent1}$
$||X_i|Waitfor1||_{Agent2}$
$||X_j|Waitfor2||_{Agent3}$

Fake signals are used to conceal the true plant state.
$||X_k{}'.Send(u)|Conceal||_{Agent1}$
$||X_i|Waitfor1.x_s(s)||_{Agent2}$
$||X_j|Waitfor2||_{Agent3}$

The next agent opens the valve and signals the third agent.
$||X_i|OpenValve||_{Agent2}$
$||X_j|Waitfor2||_{Agent3}$
$||X_j|CloseValve||_{Agent3}$

The attack completes.

Figure 4. Concealing the coordinated attack.

to mark the route. Each node may also probabilistically forward a message copy to "observer" agents for comparison.

The attributes provide a formal definition for observer agents, which use the information to make state determinations. Note that in [9], IP traceback algorithms were used to detecting the locations of malicious agents – a different goal from the one addressed in this work. Here, the observer algorithm uses

$$
\begin{array}{l}
Observe := x_{Ob_j}(z) + [z \in M]UpdateState + [z \in C]UpdatePath \\
\\
UpdateState := \nu p\,(Store(z, STORE) \supset STORE' \\
\quad + \sum_i [z \in C_i \wedge \neg(Marked(z.path)]Store(z, STATE) \supset STATE' \\
\quad + ([p \leqslant rand()]EvaluateState \oplus Observe) \\
\\
EvaluateState := \ (Evaluate(STATE, \tilde{c}) \supset CRITICAL' \\
\quad + [CRITICAL]Alert) + Observe \\
\\
UpdatePath := Compare(u, z, \tilde{k}, STORE) \supset w \\
\quad \sum_i [\neg w]MarkPath(u, z, C_iTREE) \supset C_iTREE' \\
\quad + \sum_i [w \wedge Marked(z.path)]UnMarkPath(u, z, C_iTREE) \supset C_iTREE' + Observe \\
\\
Alert := \nu f(\perp) \quad (\bar{x}_{Op}\langle f\rangle_{Op}) \\
\quad \nu \tilde{k}\tilde{c}, STORE, STATE, CRITICAL, C_iTREE \quad || \bullet Observe||
\end{array}
$$

Figure 5. Formal specification of an observer agent.

messages and copies of messages to determine a trusted set of paths. This is demonstrated by the observer definition in Figure 5.

State determination is restricted to considering messages received on trusted paths. Figure 6 provides the initial reduction of the observer using copied messages to determine route trustworthiness. The observer receives a message and invokes the goal $UpdatePath$ to compare the message with the original. If no discrepancy is found, the observer moves to the next message. If a discrepancy is found, then it notes the route and marks the forward neighboring node as untrusted. The marked messages can be represented using a graph and defined algebraically.

The indication that the path marking algorithm responds correctly depends on whether a message is marked before or after manipulation. If the message is copied before it is manipulated, then the malicious agent node appears to deliver trustworthy messages, but any subsequent node appears untrustworthy. Hence, the next node in the communication chain is marked. Alternatively, when any previous node appears to deliver a trustworthy copy, the agent node is indicated using the bar notation. In both instances, a message traversing the agent node is not trusted for state determination.

Figure 7 demonstrates another case for examining a trustworthy message and, depending on the probability, a snapshot of state. The observer receives the original message and retains the message in $STORE$. If the message arrives on a trusted path, the message is included in $STATE$ to make a determination of the system state. If the state is detected as critical, the operator is signaled by the *Alert* goal.

Dynamic behavior such as agent migration can be accommodated. For example, a change of status for X_8 and X_6 can be represented as shown in Figure 8.

$||X_i.Send||\quad (i = 1, 2, 3)$
$||Observe.x_{ob_j}(z)||_{Observer_j}\quad (z \in C)$
$||UpdatePath.Compare(u, z, \tilde{k}, STORE) \supset w||_{Observer_j}$

Case 1 - No discrepancy, message on unmarked path
$||Observe||_{Observer_j}$

Case 2 - No discrepancy, message on previously marked path
Update paths
$||UpdatePath.UnMarkPath(u, z, C_iTREE) \supset C_iTREE'||_{Observer_j}$
$\{w = TRUE \wedge Marked(z.path)\}$

Case 3 - Discrepancy found
Update paths
$||UpdatePath.MarkPath(u, z, C_iTREE) \supset C_iTREE'||_{Observer_j}$
$(w \neq TRUE)$

Figure 6. Using copied messages to determine route trustworthiness.

$||X_i.Send||\quad (i = 1, 2, 3)$
$||Observe.x_{ob_j}(z)||_{Observer_j}\quad (z \in M)$
$||UpdateState.Store(z, STORE) \supset STORE'||_{Observer_j}$

Trusted messages are stored in STATE variable
$||Store(z, STATE)||_{Observer_j}$
$\rightarrow\quad (p > rand())$
$||Observe||_{Observer_j}$
$\rightarrow\ (p \leqslant rand())$

Evaluate plant state using trusted messages
$||EvaluateState.Evaluate(STATE, \tilde{c}) \supset CRITICAL'||_{Observer_j}$

No critical state found, continue
$||Observe||_{Observer_j}$

Alert on finding a critical state
$||Alert||_{Observer_j}$

Figure 7. Using store and alert messages to determine trustworthiness.

The ability to track the possible range of dynamic system behavior is a key aspect of the modeling technique. This provides the ability to consider probable outcomes of any state determination during a changeover in node state.

The ability to represent dynamic defense strategies facilitates reasoning about agent interaction schemes. Indeed, the modeling approach facilitates

$||X_{10}.Mark.\bar{x}\langle y\rangle_{Op}|X_{10}.x_{10}(z)|C_1.\bar{x}_{12}\langle z\rangle_{Op}|X_{12}.x_{12}(z)||_{System}$

$||X_{10}.Mark.\bar{x}_8\langle y\rangle_{Op}|X_{12}.Mark.\bar{x}_9\langle y\rangle_{Op}|$
$X_9.Mark.x_9(z)|C_1.\bar{x}_{13}\langle y\rangle_{Op}|X_{13}.x_{13}(z)||_{System}$
$||X_8'.x_8(z)||_{Agent2}$

$||X_8'.Mark.\bar{x}_5\langle y\rangle_{Op}||_{Agent2}$
$||X_{13}.Mark.\bar{x}_{10}\langle y\rangle_{Op}|X_9.Mark.Observe(y)|$
$C_1.\bar{x}_{12}\langle z\rangle_{Op}|X_{12}.x_{12}(z)|X_{10}.x_{10}(z)|X_5.x_5(z)||_{System}$

$||X_{10}.Mark.\bar{x}_7\langle y\rangle_{Op}|X_5.Mark.\bar{x}_2\langle y\rangle_{Op}|$
$X_{12}.Mark.\bar{x}_9\langle y\rangle_{Op}|X_9.Mark.\bar{x}_6\langle y\rangle_{Op}|$
$C_1.\bar{x}_{13}\langle y\rangle||_{System}$

Case 1 -X_8 and X_6 are marked
$\coprod_i ||UpdatePath.MarkPath(c, y, C_1TREE)||_{Ob_i}$

Case 2 - Neither node is marked

Case 3 - X_6 is marked but not X_8

Figure 8. Dynamic behavior of agent migration.

the analysis of agent behavior in order to determine the appropriate responses during a coordinated attack.

6. Conclusions

The π-Calculus extension described in this paper uses goal-based syntax and semantics to explicitly capture the operation of agents in critical infrastructure environments. It also provides the ability to model an increased range of attack and defense capabilities compared with previous approaches. Specifically, it facilitates the modeling of coordinated attacks and defenses, and the ability to reason about complex interactions at a granular level. The example scenario demonstrates state determination in the face of a coordinated attack by leveraging trusted paths.

Our future work will concentrate on modeling and analyzing complex operator and adversary interactions in critical infrastructure environments. We will also seek to extend the approach to incorporate learning behavior and timing considerations.

References

[1] S. Boyer, *Supervisory Control and Data Acquisition*, ISA, Research Triangle Park, North Carolina, 2010.

[2] S. Braynov and M. Jadliwala, Representation and analysis of coordinated attacks, *Proceedings of the ACM Workshop on Formal Methods in Security Engineering*, pp. 43–51, 2003.

[3] S. Braynov and M. Jadliwala, Detecting malicious groups of agents, *Proceedings of the First IEEE Symposium on Multi-Agent Security and Survivability*, pp. 90–99, 2004.

[4] T. Chen, Stuxnet, the real start of cyber warfare? *IEEE Network*, vol. 24(6), pp. 2–3, 2010.

[5] B. Genge and C. Siaterlis, Investigating the effect of network parameters on coordinated cyber attacks against a simulated power plant, *Proceedings of the Sixth International Workshop on Critical Information Infrastructure Security*, 2011.

[6] R. Krutz, *Securing SCADA Systems*, Wiley, Indianapolis, Indiana, 2006.

[7] T. McEvoy and S. Wolthusen, A formal adversary capability model for SCADA environments, *Proceedings of the Fifth International Workshop on Critical Information Infrastructure Security*, pp. 93–103, 2010.

[8] T. McEvoy and S. Wolthusen, A plant-wide industrial process control security problem, in *Critical Infrastructure Protection V*, J. Butts and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 47–56, 2011.

[9] T. McEvoy and S. Wolthusen, Defeating node-based attacks on SCADA systems using probabilistic packet observation, *Proceedings of the Sixth International Workshop on Critical Information Infrastructure Security*, 2011.

[10] S. McLaughlin, On dynamic malware payloads aimed at programmable logic controllers, *Proceedings of the Sixth USENIX Conference on Hot Topics in Security*, p. 10, 2011.

[11] C. Patsakis and N. Alexandris, New malicious agents and SK virii, *Proceedings of the International Multi-Conference on Computing in the Global Information Technology*, p. 29, 2007.

[12] D. Sangiorgi and D. Walker, *π-Calculus: A theory of mobile processes*, Cambridge University Press, Cambridge, United Kingdom, 2001.

[13] D. Song and A. Perrig, Advanced and authenticated marking schemes for IP traceback, *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 878–886, 2001.

[14] G. Tesauro, D. Chess, W. Walsh, R. Das, A. Segal, I. Whalley, J. Kephart and S. White, A multi-agent systems approach to autonomic computing, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 464–471, 2004.

[15] M. Wooldridge, *An Introduction to MultiAgent Systems*, Wiley, Chichester, United Kingdom, 2002.

III

INFRASTRUCTURE SECURITY

Chapter 9

INFRASTRUCTURE PROTECTION IN THE DUTCH FINANCIAL SECTOR

Matthijs van Oers, Leon Strous, and Ron Berndsen

Abstract This paper presents a case study of critical infrastructure protection in the Dutch financial sector. The organizational structures are examined to discern the roles and functions that facilitate public-private cooperation. An assessment of the organizational structures is provided along with a description of how key organizations are identified. Finally, a basic model is presented that can be used by other sectors as a template for determining the appropriate organizational structures for critical infrastructure protection.

Keywords: Financial sector, protection, payments and securities systems

1. Introduction

After the September 11, 2001 terrorist attacks in the United States, the Dutch Government sent a letter [7] to the Dutch Parliament that included an action plan for anti-terrorism and safety [6]. This plan formed the basis for critical infrastructure protection efforts in The Netherlands and resulted in a detailed 2005 report [8] that described the critical sectors, their vulnerabilities, existing protective measures, new measures and follow-up actions. This report was used as the foundation for sector-specific efforts on critical infrastructure protection, including efforts in the financial sector.

Although critical infrastructure protection efforts initially focused on anti-terrorism and safety, the goal of critical infrastructure protection has evolved to include resilience in the face of disasters and other events as well as mitigating their risk and impact. For example, the threat of flooding, which is highly relevant in The Netherlands, is considered to lie within the scope of critical infrastructure protection.

Critical infrastructure protection is an essential activity for public and private entities. Issues that need to be addressed in a successful critical infrastructure protection approach are: (i) scope of protection; (ii) appropriate or-

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 113–126, 2012.

Figure 1. Financial sector protection concepts.

ganizational structures; (iii) required levels of protection; and (iv) measures required to achieve the required levels of protection. Solutions that address these issues are by no means straightforward and different solutions exist for different sectors and different countries.

This paper presents a case study of critical infrastructure protection in the Dutch financial sector. The organizational structures for critical infrastructure protection in The Netherlands are provided along with an assessment of their effectiveness. A general model for critical infrastructure protection that is applicable to other sectors is also presented. Note that the emphasis is on sector-specific issues, not on the more general functions of government (e.g., disaster and crisis management by police forces and other emergency services).

2. Dutch Financial Sector Approach

This section provides an overview of the critical infrastructure protection approach adopted by the Dutch financial sector.

2.1 Sector-Specific Protection

The first phase of critical infrastructure protection efforts in The Netherlands occurred from 2001 to 2005. The government and stakeholders, including The Netherlands Bank (DNB) (also known as the Dutch Central Bank) produced a report [8] that defined and identified: (i) the critical infrastructure as a whole; (ii) critical sectors; (iii) critical products and services in the critical sectors; and (iv) critical points.

Included in the report were major risks and vulnerabilities with regard to the financial sector (e.g., terrorism, natural hazards and cyber crime). Note that financial risk is not within the scope of critical infrastructure protection – the financial risk of individual institutions is specifically addressed by prudential supervisors whereas overall financial stability is primarily the responsibility of the central bank. The Netherlands Bank introduced the concept of the financial core infrastructure (FCI), which comprises the most important institutions in the Dutch financial sector. Figure 1 highlights the main concepts related to financial sector protection in The Netherlands from the broadest to the most specific.

The Dutch critical infrastructure encompasses the sectors that, if disrupted, could have a serious impact in terms of human casualties, economic losses and/or societal upheaval. Within the financial sector, payments and securities are identified as critical products and services. The critical points are defined

as the buildings, installations, systems and geographical regions that are necessary for delivering the critical products and services. Note that critical points, which include large data centers that provide services to financial institutions, are not necessarily owned by FCI institutions nor are they under financial regulation. For security reasons, the list of critical points is not publicized by the government.

2.2 Critical Products and Services

A financial infrastructure protection working group established in 2005 identified payments and securities as critical products and services in the Dutch financial sector. Although a disruption of the payments and securities infrastructure would not directly lead to human casualties, a major outage could have serious consequences, including societal upheaval.

The payments and securities domain can be categorized as: (i) retail payments; (ii) wholesale payments; and (iii) securities. The distinction is intended to emphasize the differences with respect to products, customers, institutions and regulators.

- **Retail Payments Domain:** This domain consists of payment systems, products and services for consumers and corporations, along with the accompanying infrastructures. Examples of products are debit cards, credit cards, money transfers, cash payments and direct debits. Institutions involved in processing retail products are banks, automated clearing houses (ACHs) and payment settlement infrastructures.

- **Wholesale Payments Domain:** This domain consists of the inter-bank payment infrastructures and actors involved in large value (low volume) payments, foreign exchange and other money market products. Institutions involved in processing wholesale payments include operators of settlement systems, banks and institutions that conduct foreign exchange transactions (e.g., CLS).

- **Securities Domain:** This domain consists of trading platforms for equities and derivatives and the accompanying clearing and settlement infrastructures along with their various actors. The institutions include exchanges (e.g., NYSE Euronext) and clearing and settlement infrastructures (e.g., LCH Clearnet, EMCF and Euroclear).

Note that the payments and securities infrastructure is international in its scope and is very reliant on information and communications technology. Indeed, many financial organizations deliver cross-border services to a multitude of customers. Interested readers are referred to [4] for a detailed description of the various products and services at the European level.

2.3 Dutch Financial Infrastructure

The 2005 financial infrastructure protection working group established an initial list of critical institutions. However, the working group did not develop a formal method to evaluate institutions on a recurring basis. In the following, we describe the method for listing an institution as an FCI.

The Netherlands Bank is responsible for compiling the FCI list in collaboration with the Ministry of Finance and the Netherlands Authority for Financial Markets. The following qualitative criteria are used in the determination:

- Disruption of the institution leads to large economic losses or large-scale civil unrest.
- The institution is directly supervised and regulated by the appropriate Dutch authorities, namely The Netherlands Bank and the Netherlands Authority for Financial Markets.

Institutions are added to the FCI list if their total transaction volume or value is in the top 80% of all financial institutions. The application of the criteria identified six institutions in the retail payments domain, five in the wholesale payments domain and seven in the securities domain, yielding a total of fourteen institutions in the Dutch FCI list (some institutions are listed in more than one domain). Note, however, that The Netherlands Bank has the discretionary power to add or remove an institution if special circumstances warrant.

An organization identified as an FCI is susceptible to the following additional regulatory requirements:

- Compliance with The Netherlands Bank Business Continuity Assessment Framework [9].
- Participation in the financial sector's Crisis Management Organization to address operational disruptions of the payments and securities infrastructure.
- Participation in the Dutch terrorism alert system.
- Participation in meetings of the Business Continuity Platform for Critical Infrastructure Protection.
- Participation in market-wide simulation exercises.

Critical points can be, but are not necessarily part of, institutions in the FCI list. The identification of critical points is primarily the responsibility of FCI institutions as part of their regular risk managements and business continuity processes. Note that specific arrangements are made for critical points that are deemed essential to the entire financial sector (e.g., Swift, which provides secure financial messaging services).

2.4 International Context

The starting point for critical infrastructure protection is often a nationally-driven program. However, many financial market infrastructures operate across international borders. Indeed, regulation and oversight are often considered per institution and not per country. In the financial sector, oversight is a central bank function whose goal is to mitigate systemic risk while contributing to the smooth operation of the payments system. For example, crisis management of Target2, the European real-time gross settlement system for inter-bank payments, is led by the European Central Bank in collaboration with the other Eurosystem central banks.

In the European context, critical infrastructure protection in the financial sector focuses on the most important institutions in the European Union. This set includes all the institutions that have been identified by their home countries as critical.

2.5 Organizational Structures and Measures

After defining the scope of protection, the next step is to determine the organizational structures, adequate level(s) of protection and appropriate protection measures. In most cases, organizations leverage structures that are already in place and modify them as required to incorporate critical infrastructure protection tasks. Developing the right organizational structures for critical infrastructure protection within a sector, however, does present some challenges. Commercial parties are driven by profit and are not always prepared to invest in projects that add costs. Additionally, organizations are reluctant to share information necessary for critical infrastructure protection efforts to external entities, especially competitors. Overcoming these challenges demands a government authority or regulator to take a lead role. Also, the organizational structures should strike a balance between the demand for resources and the ability to obtain tangible results.

The development of measures for protecting critical infrastructures typically draws on experience, relevant events and historical data. This approach, however, is not well suited to dealing with events that manifest themselves only a few times in history such as a pandemic or the September 11, 2001 terrorist attacks. Indeed, the Fukushima nuclear disaster in 2011 demonstrates that historical information does not provide adequate guidance for protecting against unexpected events.

The success of a critical infrastructure protection approach is strongly influenced by the organizational structures and protective measures. The organizational structures can be categorized as: (i) public; (ii) public-private; and (iii) private. The protective measures can be divided into two types: (i) preventative measures that increase the resilience of critical processes; and (ii) corrective or responsive measures that decrease the impact of a crisis.

The Dutch financial sector engages a mixture of organizational structures and measures to enhance FCI resilience (Table 1). Note that this paper focuses

Table 1. Summary of national organizational structures and measures.

Structure	Category	Measures
Public	**Preventative** Prudential Supervision; Oversight; The Netherlands Bank	**Preventative** Business Continuity Assessment Framework; Supervisory Standards
	Ministry of Finance; Intelligence Agencies	Policy Reports; Threat and Vulnerability Analysis
Public-Private	**Preventative** Business Continuity Platform	**Preventative** Information Sharing Best Practices Consultation on Standards
	Terrorism Alert Working Group	Anti-Terrorism Measures; Terrorism Alert System
	Cross-Sector Collaboration	Cross-Sector Exercises
	FI-ISAC	Cyber Crime Data Exchange
	Sector Crisis Management	Market-Wide Simulation Exercises
	Corrective/Responsive Sector Crisis Management	**Corrective/Responsive** Crisis Management Decision-Making and Communication; Disaster Recovery Planning
Private	None	None

on the financial sector; therefore, other actors (e.g., Ministry of Interior and Ministry of Justice and Security) are not included because their relevance to the organization and implementation of critical infrastructure protection is not sector-specific. The same is true for protection measures such as the organization of special anti-terrorism police forces and the strategic stockpiling of oil and diesel in case of shortages.

Public Structures. The financial sector is subject to many regulators and policy makers. The most relevant global entities are the Bank for International Settlements (BIS), Financial Stability Board (FSB) and International Organization of Securities Commissions (IOSCO). At the European level, the entities include the European System of Central Banks (ESCB), European Banking Authority (EBA) and European Securities Markets Authority (ESMA). The Dutch entities include The Netherlands Bank (DNB), Ministry of Finance and Netherlands Authority for Financial Markets (AFM).

Although regulators and policy makers have different scopes, objectives and approaches, all of them have the common goal of financial sector stability. The institutions and organizational structures issue standards (e.g., guidelines, rec-

ommendations, principles and expectations) and/or perform supervision and oversight. The supervisors and overseers regularly assess the operational reliability, security and business continuity against the standards.

FCI institutions must comply with the requirements of The Netherlands Bank Business Continuity Management Assessment Framework [9]. These principle-based requirements address several areas: strategy and policy, business impact analysis and risk analysis, scenarios and measures, testing and monitoring, management and maintenance, and crisis management and communications. The principal-based requirements leave options for institutions to develop their own solutions, unlike rule-based requirements that prescribe exactly what must be implemented.

Additionally, the Ministry of Finance, The Netherlands Bank and Dutch intelligence community collaborate closely on critical infrastructure protection initiatives. These entities develop policy reports and perform threat analyses on a regular basis.

Public-Private Structures. There are two main reasons for establishing public-private partnerships for critical infrastructure protection. The first is that critical infrastructures incorporate both private and public investments; protecting these infrastructures is the task of the central government and requires collaboration with private sector asset owners and operators. The second reason is that public-private partnerships facilitate the management of cross-sector dependencies. The financial sector, for example, is heavily dependent on the telecommunications and energy sectors. Cooperation is required in order to optimize the level of protection. Cross-sector cooperation can also occur in private partnerships, but experience has shown that some form of public interaction or initiating force is key to success.

The following public-private partnerships have been instituted in the Dutch financial sector:

- **Business Continuity Platform for the Critical Infrastructure Financial Sector (BC-CIF):** The Netherlands Bank initiated and currently chairs this platform whose goal is to share knowledge and best practices on business continuity and crisis management between FCI institutions and with the Ministry of Finance. The platform serves as a coordination point for the financial sector with regard to governmental critical infrastructure protection initiatives. Examples of the shared information are best practices related to outsourcing of critical processes and vendor requirements.

- **Working Group on Alerting to Terrorism in the Financial Sector (WAFS):** This working group was created to facilitate the exchange of information on terrorism threats, anti-terrorism measures and the terrorism alert system. The Netherlands Bank chairs the working group, which includes FCI institutions, intelligence agencies and the Ministry of Finance. The working group has developed and implemented anti-

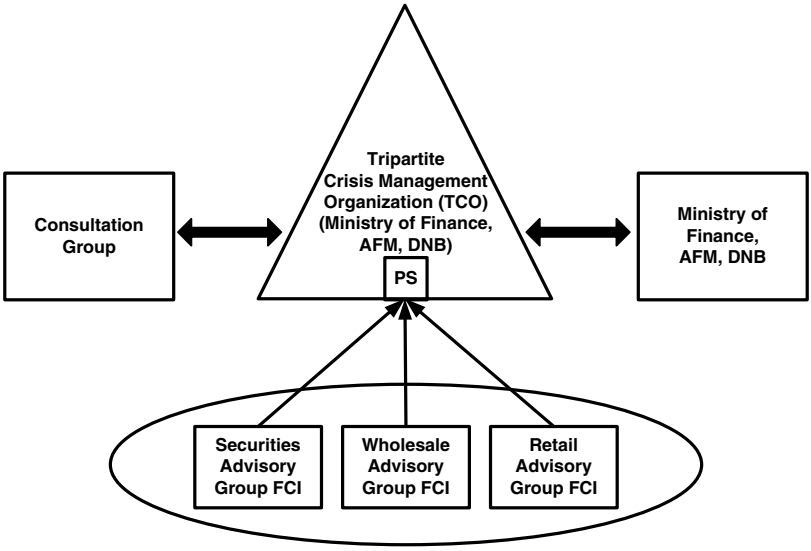


Figure 2. Crisis management structure.

terrorism measures that are activated depending on the threat level. The working group entities are connected to the terrorism alert system (organized, funded and operated by the government), which is designed to alert institutions in the critical infrastructure sectors to terrorist threats. Every critical sector in the Netherlands is connected to this system. Currently, the terrorism alert system is designed to warn of physical threats; however, efforts are underway to extend the system to include cyber threats.

- **Financial Institutions Information Sharing and Analysis Centre (FI-ISAC):** The goal of FI-ISAC is to exchange information between banks, infrastructures and government organizations in order to prevent and respond to cyber security incidents that could lead to fraud, loss of reputation and other risks. The FI-SAC works closely with the National Cyber Security Centre (NCSC).

- **Sector Crisis Management:** This sector-level structure is designed to perform corrective and responsive actions in the event of a major disruption to the payments and securities infrastructure. The structure comprises the Tripartite Crisis Management Organization (TCO), which incorporates the Ministry of Finance, Netherlands Authority for Financial Markets and The Netherlands Bank as board-level entities. A Consultation Group consisting of board members of FCI institutions, and various Advisory Groups provide recommendations to the TCO. The TCO is supported by a Permanent Secretariat (PS) that helps manage collaboration. Figure 2 presents the crisis management structure for the payments and securities infrastructure.

Table 2. Cross-border organizational structures and measures.

Structure	Category	Measures
Public	**Preventative** Oversight	**Preventative** CPSS-IOSCO Principles for Financial Market Infrastructures; Crisis Management Exercises
	Corrective/Responsive Eurosystem Crisis Management	**Corrective/Responsive** Crisis Management Decision-Making and Communication
Public-Private	None	None
Private	None	None

- **Cross-Sector Cooperation with the Telecommunications Sector:** In order to manage the effects of interdependencies in the financial sector, Platform BC-CIF started a collaboration with the National Continuity Forum for the Telecommunication Sector. An example of this collaboration is a jointly-organized crisis management exercise that seeks to strengthen cross-sector resilience.

Private Structures. No specific private partnerships related to critical infrastructure protection currently exist in the Dutch financial sector. However, a few structures have been created that indirectly support critical infrastructure protection goals. An example is a task force organized by the Dutch Bankers Association to address cyber crime threats.

2.6 Cross-Border Structures and Measures

Nationally-oriented critical infrastructure protection is limited because the majority of the financial market institutions operate across national borders. Indeed, critical infrastructure protection in the financial sector is quite complex with regard to coordination, legal aspects, ambiguities of roles and responsibilities, and vulnerabilities. Currently, the only cross-border collaborations that exist are public-only partnerships involving the European System of Central Banks and the Bank of International Settlements (BIS). These entities coordinate oversight, standard setting and crisis management activities across the Eurozone, European Union as well as globally. An example standard is the CPSS-IOSCO Principles for Financial Market Infrastructures [2].

Table 2 summarizes the cross-border collaborations and structures related to critical infrastructure protection in the financial sector. Clearly, cross-border coordination is still in its infancy and is an area that needs improvement.

2.7 Challenges

In general, critical infrastructure sectors are relatively easy to identify (e.g., energy, telecommunications, finance and health care). However, in some sectors, it is difficult to identify the critical services and processes, institutions and components. This is often the case in a highly networked infrastructure where small components can be essential to the overall function.

A large number of critical components in a sector can render it difficult to manage effectively. Alternatively, a small number of critical components can induce neglect and complacency with regard to overall critical infrastructure protection efforts. It is important to recognize these situations and strike the right balance when prioritizing assets.

In the case of a major disaster that impacts multiple sectors such as energy and telecommunications, services from the various sectors tend to recover at different rates. This may create serious problems when one sector is dependent on another. Due to resource limitations, sectors must set priorities according to the most critical societal functions and contractual obligations.

To increase the resilience of the critical infrastructure, it is recommended to maintain the transparency of priorities to the extent possible. Transparency facilitates preparatory efforts that lessen the impact of a disaster and helps clarify the lines of responsibility of public and private sector entities.

Finally, critical infrastructures are becoming more complex, more interconnected and, in many cases, they extend beyond national borders. These developments increase the difficulty in defining organizational structures for critical infrastructure protection. Indeed, there is an urgent need to address this issue going forward.

3. Analysis

Significant critical infrastructure protection efforts have been undertaken in the Dutch financial sector. The question is whether these efforts have resulted in effective organizational structures for critical infrastructure protection.

Assaf [1] has shown that intervention with regard to critical infrastructure protection efforts ranges from pure state provisions to pure market-driven provisions. The types of intervention are identified as: command and control, delegation to agency, delegation to agency plus negotiations, enforced self-regulation and voluntary self-regulation.

The choice of the level of intervention is based on the distinction between two regulatory models for critical infrastructure protection, the national security model and the business continuity model. The national security model focuses on security and public safety, and leads to critical infrastructure protection with a preference for government intervention. The business continuity perspective is based on neoliberal economic values. Business continuity is viewed in terms of return on investment and risk management; thus, the model results in a preference for market provisions. Although the two models may align in extreme cases, they have competing sets of values that result in differ-

ent regulatory interventions. Also, as mentioned above, the differences in goals and approaches between the private sector and the public sector can also be explained from an externality perspective.

A hybrid critical infrastructure protection approach is implemented in the Dutch financial sector. For some critical infrastructure protection functions, strong government intervention exists (e.g., supervision and oversight based on legal provisions). For other functions, voluntary self-regulation exists (e.g., determining business continuity best practices and vendor requirements). This hybrid approach also addresses accountability and transparency associated with critical infrastructure protection efforts. If a high degree of accountability is needed, strict government intervention must exist; self-regulation is appropriate if trust is the most important component of the public-private partnership.

The validity of the hybrid critical infrastructure protection approach adopted by the Dutch financial sector is further strengthened by Dunn-Cavelty and Suter [3]. They argue that public-private partnerships in which the public parties have a strong role are not always optimal. Indeed, information sharing is considered to be the most important requirement for critical infrastructure protection. Information sharing requires complementary goals, mutual trust, clear distribution of risks, clear sharing of responsibilities and authority, and market- and success-oriented thinking [5]. Because of concerns related to confidential information and the divergent goals of national security versus business continuity, a strong government role may hinder effective information sharing in some critical infrastructure protection functions. Indeed, an approach where the government takes on a "meta role" is sometimes required. In such a scenario, the government is not focused on monitoring the collaborating organizations, but instead coordinates and stimulates functional networks so that the organizations can fulfill the tasks required by the state.

In the Dutch financial sector, The Netherlands Bank assumes the coordination and stimulation roles for several tasks (e.g., Platform BC-CIF). Meanwhile, the financial sector uses FI-ISAC for sharing information related to cyber security. Thus, for aspects that require a national security model, a more government-interventionist organization has been chosen by the Dutch financial sector. On the other hand, for business continuity, where information sharing is key, a low-interventionist, public-private partnership model has been selected.

4. Proposed Model

Our model for determining organizational structures for critical infrastructure protection is derived from the Dutch financial sector efforts described above. The model, which is illustrated in Figure 3, incorporates three steps:

- **Define:** The initial step in a critical infrastructure protection program is to define the scope of protection, critical processes, products and services. Additionally, a global risk analysis must be performed to identify the major vulnerability concerns. This step is project-based and requires the collaboration of public and private sector entities.

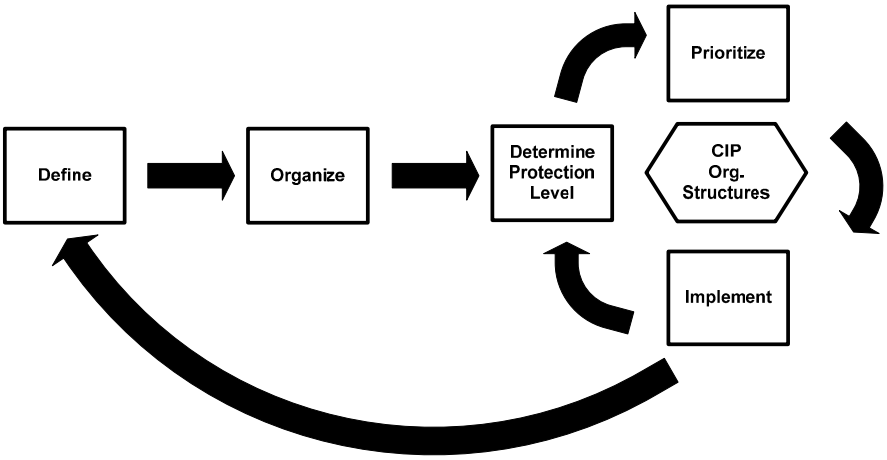


Figure 3. Model for determining organizational structures.

- **Organize:** After the definition step, it is necessary to set up the organizational structure. The organization of public-only and public-private partnerships in a sector should cover four themes: (i) anti-terrorism; (ii) business continuity based on an all-hazard approach; (iii) information and communications technology; and (iv) crisis management. These themes typically cannot be addressed by one organization because they require different decision-making mandates and/or expertise. It is, however, important to manage the intersection of the themes. The institutions that are in charge of these organizational structures should be aware of topics that cross multiple themes and a model of collaboration should be considered, ranging from pure government intervention to pure self-regulation. After these organizational structures are established, efforts to protect the infrastructure can proceed.

- **Determine Protection Level, Prioritize and Implement:** The next phase involves determining the protection levels and priorities and proceeding with the implementation. These tasks are executed by dedicated critical infrastructure protection structures (hexagon in Figure 3).

  During the determination of protection levels, preventative, corrective and response measures are also identified based on risk analysis. Following this, the priorities for implementing protection measures are determined. The setting of priorities is often influenced by statistical information about events, threats and risks, cost-benefit tradeoffs, political and societal influences, and the latest crisis. After the priorities are set, the FCI institutions implement the required protection measures.

  It is important to note that the latest crisis invariably exerts an influence on critical infrastructure protection efforts. After the attacks of Septem-

ber 11, 2001, anti-terrorism efforts were increased. When the Mexican Flu broke out, The Netherlands took strong efforts to protect its citizens from the pandemic threat. The current focus is protecting the critical infrastructure from cyber attacks. These efforts are important and our intention is not to imply that they have received too much attention. Rather, we highlight this issue because it is important not to become myopic and dismiss other potential risks.

The proposed model is intended to be applied as a cyclical feedback loop. Changes within a step can propagate to affect subsequent steps. Therefore, it is essential to perform periodic reviews and updates to account for changes in the infrastructure and threat landscape.

The participation of at least one institution (e.g., regulator, government agency or private entity) that takes the lead in organizing the initial phase of a public-private partnership is a requirement. The institution should focus first on initiating collaboration in the sector and addressing the major concerns. After the initial coordination, a tailored approach for determining the appropriate public-private partnership can be developed.

5. Conclusions

The Dutch financial sector provides a concrete example of a sector-wide approach for critical infrastructure protection. The measures implemented by individual institutions along with sector-wide efforts appear to be very effective for safeguarding critical assets. The appropriate use of public-private relationships has fostered communication and information exchange, as well as the protection of sensitive information where necessary. Government intervention has been selected for functions in which national security is the primary consideration. On the other hand, a market-oriented approach is employed for functions that rely on sharing and trust. The basic model derived from the Dutch financial sector can be used as a template by other sectors – or other countries – to determine the organizational structures that can achieve effective critical infrastructure protection.

Our future research will conduct further analysis of critical infrastructure protection in the Dutch financial sector and refine the model as appropriate. Also, it will attempt to model and analyze cross-sector and international collaborative activities related to critical infrastructure protection.

References

[1] D. Assaf, Models of critical information infrastructure protection, *International Journal of Critical Infrastructure Protection*, vol. 1, pp. 6–14, 2008.

[2] Committee on Payment and Settlement Systems, Payment and Securities Principles for Financial Market Infrastructures, Bank for International Settlements, Basel, Switzerland (`www.bis.org/publ/cpss101a.pdf`), 2011.

[3] M. Dunn-Cavelty and M. Suter, Public-private partnerships are no silver bullet: An expanded governance model for critical infrastructure protection, *International Journal of Critical Infrastructure Protection*, vol. 2(4), pp. 179–187, 2009.

[4] European Central Bank, The Payment System, Frankfurt, Germany (www.ecb.int/pub/pdf/other/paymentsystem201009en.pdf), 2010.

[5] V. Kouwenhoven, Public-private partnership: A model for the management of public-private cooperation, in *Modern Governance: New Government-Society Interactions*, J. Kooiman (Ed.), Sage, London, United Kingdom, pp. 119–130, 1993.

[6] Ministry of General Affairs, Actieplan Terrorismebestrijding en Veiligheid (in Dutch), The Hague, The Netherlands (zoek.officielebekendmakingen.nl/kst-27925-21.html), 2001.

[7] Ministry of General Affairs, Brief van de Minister-President, Minister van Algemene Zaken en van de Ministers van Justitie, van Binnen-Landse Zaken en Koninkrijksrelaties, Terroristische Aanslagen in de Verenigde Staten, Kamerstuk 27925, Nr. 10, Vergaderjaar 2001-2002 (in Dutch), The Hague, The Netherlands (zoek.officielebekendmakingen.nl/kst-27925-21.html), 2001.

[8] Ministry of the Interior and Kingdom Relations, Rapport Bescherming Vitale Infrastructuur (in Dutch), The Hague, The Netherlands (www.eerstekamer.nl/9370000/1/j9vviasdcklgjqj/vh4gfzjj2vqo/f=/vh4gfzjj2vqo.pdf), 2005.

[9] The Netherlands Bank, Assessment Framework for Financial Core Infrastructure, Business Continuity Management, Amsterdam, The Netherlands (www.dnb.nl/en/binaries/DNB\%20Assessment\%20Framework\%20Business\%20Continuity\%20version\%202011_tcm47-253700.PDF), 2011.

Chapter 10

PRIVACY-PRESERVING POWER USAGE CONTROL IN THE SMART GRID

Chun Hu, Wei Jiang, and Bruce McMillin

Abstract In the smart grid, the power usage of households are recorded and analyzed in (near) real time by utility companies. The usage data enables a utility to manage its electric power supply to neighborhoods more efficiently and effectively. For instance, to prevent a power outage during a peak demand period, the utility can determine the power supply threshold for a neighborhood. When the total power usage of the neighborhood exceeds the threshold, certain households in the neighborhood are required to reduce their energy consumption. This type of power usage control benefits electric utilities and their consumers. However, the energy usage data collected by a utility can also be used to profile an individual's daily activities – a potentially serious breach of personal privacy. To address the problem, this paper specifies distributed, privacy-preserving energy usage control protocols that enable utilities to efficiently manage power distribution while ensuring that individual power usage data is not revealed.

Keywords: Smart grid, power usage control, privacy preservation

1. Introduction

The smart grid provides utilities and consumers with intelligent and efficient ways to manage electric power usage. To achieve this, the grid needs to collect a variety of data related to energy distribution and usage. This expanded data collection raises many privacy concerns, especially with regard to energy consumers. For example, specific appliances can be identified through their electricity usage signatures from data collected by automated meters (at a frequency much higher than the traditional monthly meter readings) [11]. Indeed, research has shown that the analysis of aggregate household energy consumption data over fifteen-minute intervals can determine the usage patterns of most

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 127–137, 2012.

major home appliances [4, 10]. This increases the likelihood of discovering potentially sensitive information about consumer behavior and so-called activities of daily life (ADL) [12].

Since ADL data is generally personal or private, it should be protected from access by unauthorized entities. For example, a malicious entity could analyze the usage patterns of household appliances in energy usage data, and determine when the victim is not home. The malicious entity could then plan and initiate actions without being easily exposed.

A common strategy to prevent power outages is to dynamically adjust the power consumed by households and businesses during peak demand periods. In this case, a utility may determine a threshold for each neighborhood it services. When the total power usage by a neighborhood exceeds the threshold, some households in the neighborhood are required to reduce their energy consumption based on contractual agreements with the utility.

Implementing threshold-based power usage control (TPUC) requires a utility to collect and analyze power usage data from every household in the participating neighborhoods. Consumers are generally provided with incentives such as reduced rates to encourage participation. In return, the consumers must agree to reduce their power consumption when necessary. For example, the household that consumes the most power in a neighborhood may be required to reduce its consumption to bring the total power usage of the neighborhood under the threshold.

Privacy concerns regarding the fine-granular power usage data that is required to be collected and stored by utilities is the primary obstacle to implementing TPUC in the smart grid. To address these concerns, it is important to design sophisticated TPUC protocols that preserve the privacy of both consumers and utilities. This paper describes two distributed, privacy-preserving protocols that enable utilities to efficiently manage power distribution while satisfying the privacy constraints.

2. Problem Statement

Let $A_1, \ldots, A_n$ be n participating consumers or users from a neighborhood. Furthermore, let f_{TPUC} be a privacy-preserving TPUC protocol given by:

$$f_{\text{TPUC}}\left(\{a_1, \ldots, a_n\}, t\right) \rightarrow \left(\{\delta_1, \ldots, \delta_n\}, \perp\right)$$

where $a_1, \ldots, a_n$ are the average power consumptions during a fixed time interval by consumers $A_1, \ldots, A_n$, respectively; and t is a threshold determined by the utility for the neighborhood. The protocol returns δ_i to consumer A_i and nothing to the utility. The $\delta_1, \ldots, \delta_n$ values are the required power consumption adjustments for the consumers such that $t \geq \sum_{i=1}^{n} (a_i - \delta_i)$. When $t \geq \sum_{i=1}^{n} a_i$, every δ_i is equal to zero, i.e., no power usage adjustments are required. Note that not all the consumers are required to make adjustments at a given time. In general, the specific adjustments that are made depend on the strategy agreed upon by the consumers and the utility.

This paper considers two common power adjustment strategies:

- **Maximum Power Usage:** When the average total energy consumption by a neighborhood over a fixed time interval or round (denoted by $a = \sum_{i=1}^{n} a_i$) exceeds a predefined threshold t, then the consumer who has used the most power during previous round is asked to reduce his or her power consumption. After the next round, if the new a that is computed is still greater than t, then the newly-found maximum energy consumer is asked to reduce his or her usage. This process is repeated until $t \geq a$. Note that the a value is computed at the end of each round. During each round, the consumer who has used the most power can reduce his or her consumption without much discomfort by shutting down one or more household appliances (e.g., washer and dryer) or by adjusting the thermostat temperature setting a few degrees.

- **Individual Power Usage:** If the average total energy consumption a is over the threshold t, then the consumption of every consumer in the neighborhood is reduced based on his or her last usage a_i. The least amount of energy reduction δ_i for each user A_i is determined by the following equation:

$$\delta_i = \frac{a_i}{a}(a - t) \quad \text{and} \quad a = \sum_{i=1}^{n} a_i \tag{1}$$

  where δ_i is a lower bound on the amount of power usage that the user A_i should cut, and a is the average total power usage during the last time interval. After the adjustments, the average total power usage falls below t. Thus, under this strategy, the protocol only has only one round of execution.

Since the collection of fine-granular power usage data by a utility can compromise personal privacy, it is important to prevent the disclosure of such data. Therefore, an f_{TPUC} protocol should satisfy two privacy-preserving requirements:

- **Consumer Privacy:** The average power usage data a_i of a consumer A_i should not be disclosed to any other consumer in the neighborhood or to the utility during the execution of an f_{TPUC} protocol.

- **Utility Privacy:** The threshold t should not be disclosed to the consumers of a neighborhood during the execution of an f_{TPUC} protocol.

The utility privacy requirement must be met because an entity who knows the t values for a number of neighborhoods serviced by a utility could infer the operational capacity and the energy supply distribution of the utility. The public disclosure of this information can cause the utility to lose its competitive advantage. We adopt security definitions from the domain of secure multiparty computation [14, 15] to develop the rigorous privacy-preserving TPUC protocols described in this paper.

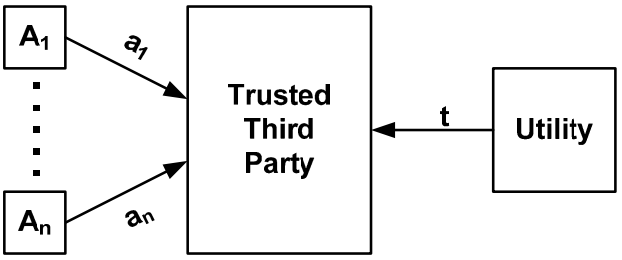


Figure 1. TTP-based f_{TPUC} protocol.

A naive – albeit secure – way to implement an f_{TPUC} protocol is to use a trusted third party (TTP). As shown in Figure 1, each consumer A_i sends his or her a_i value to a TTP while the utility sends its t value to the TTP. Having received these values, the TTP compares t with $a = \sum_{i=1}^{n}$. If $t < a$, the TTP computes each δ_i value and sends it to consumer A_i.

This TTP-based f_{TPUC} protocol easily meets the privacy-preserving requirement. However, such a TTP rarely exists in practice. Therefore, it is necessary to develop f_{TPUC} protocols that do not use a TTP while achieving a similar degree of privacy protection provided by a TTP protocol.

3. Related Work

This section briefly reviews the related work in the field. In particular, it discusses privacy issues in the smart grid, and presents key security definitions from the domain of secure multiparty computation.

Privacy issues in the smart grid are highlighted in [12]. Our work primarily focuses on one of these issues, namely, protecting the release of fine-granular energy usage data in a smart grid environment. Quinn [11] has observed that power consumption data collected at relatively long intervals (e.g., every fifteen or thirty minutes) can be used to identify the use of most major household appliances. Indeed, data collected at fifteen-minute intervals can be used to identify major home appliances with accuracy rates of more than 90 percent [10]. Furthermore, the successful identification rate is near perfect for large two-state household appliances such as dryers, refrigerators, air conditioners, water heaters and well pumps [4]. Lisovich, *et al.* [8] describe the various types of information that can be inferred from fine-granular energy usage data.

In this paper, privacy is closely related to the amount of information disclosed during the execution of a protocol. Information disclosure can be defined in several ways. We adopt the definitions from the domain of secure computation, which were first introduced by Yao [14, 15]. The definitions were subsequently extended to multiparty computation by Goldreich, *et al.* [6].

We assume that the protocol participants are "semi-honest." A semi-honest participant follows the rules of a protocol using the correct inputs. However, the participant is free to later use what he or she sees during the execution

of the protocol to compromise privacy (or security). Interested readers are referred to [5] for detailed definitions and models.

The following definition formalizes the notion of a privacy-preserving protocol with semi-honest participants.

Definition. Let T_i be the input of party i, $\prod_i(\pi)$ be i's execution image of the protocol π and s be the result computed from π. π is secure if $\prod_i(\pi)$ can be simulated from $\langle T_i, s\rangle$ and the distribution of the simulated image is computationally indistinguishable from $\prod_i(\pi)$.

Informally, a protocol is privacy-preserving if the information exchanged during its execution does not leak any knowledge regarding the private inputs of any participants.

4. Privacy-Preserving Protocols

We specify two privacy-preserving TPUC protocols: f^1_{TPUC} and f^2_{TPUC} for the maximum power usage strategy and the individual power usage strategy, respectively. We adopt the same notation as before: $A_1, \ldots, A_n$ denote n utility consumers in a participating neighborhood, and $a_1, \ldots, a_n$ denote the average power usage during a fixed time interval set by utility C. Additionally, $a = \sum_{i=1}^{n} a_i$ and $a^m \in \{a_1, \ldots, a_n\}$ denotes the maximum individual energy usage of consumer $A^m \in \{A_1, \ldots, A_n\}$. Without loss of generality, we assume that a^m is unique and $a_1, \ldots, a_n$ are integer values. Since $a_1, \ldots, a_n$ can be fractional values in the real world, the values have to be scaled up to the nearest integers before the protocols can be used. After the results are returned by the protocols, they are adjusted by the appropriate scaling factors to obtain the final values.

The privacy-preserving requirements (consumer privacy and utility privacy) described above are difficult to achieve without using a trusted third party. Consequently, we relax the privacy-preserving requirements slightly in defining the protocols. In particular, the two privacy-preserving requirements are specified as follows:

- **Maximum Power Usage:** Only a and a^m can be disclosed to $A_1, \ldots, A_n$.
- **Individual Power Usage:** Only a can be disclosed to $A_1, \ldots, A_n$.

Note that these relaxed requirements permit the design of efficient protocols.

The f^1_{TPUC} and f^2_{TPUC} protocols require several primitive protocols as subroutines. These primitive protocols are defined as follows:

- *Secure_Sum*$(a_1, \ldots, a_n) \rightarrow a$
  This protocol has n (at least three) participants. Each participant A_i has an a_i value, which is a protocol input. At the end of the protocol, a is known only to A_1.
- *Secure_Max*$(a_1, \ldots, a_n) \rightarrow a^m$
  This protocol has n participants. Each participant A_i has an a_i value,

1. A_1 randomly selects $r \in \{0, N-1\}$, computes $s_1 = a_1 + r \mod N$ and sends s_1 to A_2
2. A_i $(1 < i < n)$ receives s_{i-1}, computes $s_i = s_{i-1} + a_i \mod N$ and sends s_i to A_{i+1}
3. A_n receives s_{n-1}, computes $s_n = s_{n-1} + a_n \mod N$ and sends s_n to A_1
4. A_1 receives s_n and computes $a = s_n - r \mod N$

Figure 2. Secure_Sum protocol.

which is a protocol input. At the end of the protocol, a^m is known to every participant, but a_i is only known to A_i.

- *Secure_Compare*$(a, t) \rightarrow 1$ *if* $a > t$ *and 0 otherwise*
  This protocol has two participants. At the end of the protocol, both participants know if $a > t$.

- *Secure_Divide*$((x_1, y_1), (x_2, y_2)) \rightarrow \frac{x_1+x_2}{y_1+y_2}$
  This protocol has two participants. Participants 1 and 2 submit the private inputs (x_1, y_1) and (x_2, y_2), respectively. At the end of the protocol, both participants know $\frac{x_1+x_2}{y_1+y_2}$.

All these primitive protocols have privacy-preserving properties because the private input values are never disclosed to other participants.

4.1 Implementation

The Secure_Sum protocol can be implemented in several ways. In this paper, we adopt a randomization approach, which yields the protocol specified in Figure 2. Note that N is a very large integer. Because r is randomly chosen, s_1 is also a random value from the perspective of A_2. Therefore, A_2 is not able to discover a_1 from s_1. Following the same reasoning, $a_1, \ldots, a_n$ are never disclosed to the other consumers during the computation process. Because A_1 is the only participant who knows r, only A_1 can derive a correctly.

The remaining three primitive protocols are straightforward to implement. The Secure_Max protocol is implemented using the steps given in [13]. The Secure_Compare protocol is implemented using the generic solution given in [2]. The Secure_Divide protocol is implemented using the methods outlined in [1, 3].

4.2 f^1_{TPUC} Protocol

The f^1_{TPUC} protocol is readily implemented using the primitive protocols. Figure 3 presents the main steps in the protocol.

Since A_1 has the value a, the Secure_Compare protocol in Step 2 can only be executed between consumer A_1 and the utility. However, any consumer can become A_1; this is accomplished via a leader election process among the

1. A_1 obtains $a \leftarrow \text{Secure\_Sum}(a_1, \ldots, a_n)$
2. A_1 and the utility jointly perform the Secure_Compare protocol
   If $\text{Secure\_Compare}(a, t) = 1$, then
   (a) Each A_i obtains $a^m \leftarrow \text{Secure\_Max}(a_1, \ldots, a_n)$
   (b) A^m (self-identified via a^m) reduces his or her energy consumption
3. The above steps are repeated until $\text{Secure\_Compare}(a, t) = 0$

Figure 3. f^1_{TPUC} protocol.

consumers that determines who becomes A_1. Alternatively, A_1 can be chosen at random before each execution of the protocol.

4.3 f^2_{TPUC} Protocol

In the f^2_{TPUC} protocol, A_1 is also responsible for the Secure_Sum and Secure_Compare operations. An additive homomorphic probabilistic public key encryption (HEnc) system is used as a building block in the protocol. The private key is only known to the utility and the public key is known to all the participating consumers.

Let E_{pk} and D_{pr} be the encryption and decryption functions in an HEnc system with public key pk and private key pr. Without pr, it is not possible to discover x from $E_{pk}(x)$ in polynomial time. (Note that, when the context is clear, the subscripts pk and pr in E_{pk} and D_{pr} are omitted.) The HEnc system has the following properties:

- The encryption function is additive homomorphic, i.e., $E_{pk}(x_1) \times E_{pk}(x_2) = E_{pk}(x_1 + x_2)$.
- Given a constant c and $E_{pk}(x)$, $E_{pk}(x)^c = E_{pk}(c \cdot x)$.
- The encryption function has semantic security as defined in [7], i.e., a set of ciphertexts do not provide additional information about the plaintext to an unauthorized party or $E_{pk}(x) \neq E_{pk}(x)$ with very high probability.
- The domain and the range of the encryption system are suitable.

Any HEnc system is applicable, but in this paper, we adopt Paillier's public key homomorphic encryption system [9] due to its efficiency. Informally, the public key in the system is (g, N), where N is obtained by multiplying two large prime numbers and $g \in \mathbb{Z}^*_{N^2}$ is chosen randomly.

To implement the f^2_{TPUC} protocol and according to Equation (2), each consumer A_i needs to calculate $\frac{a_i \cdot t}{a}$ between A_i and the utility C so that a_i is not disclosed to C and t is not disclosed to A_i. We adopt the Secure_Divide primitive and an HEnc system to solve the following problem:

$$\delta_i = \frac{a_i}{a}(a - t) = a_i - \frac{a_i \cdot t}{a} \tag{2}$$

1. A_1 obtains $a \leftarrow \text{Secure\_Sum}(a_1, \ldots, a_n)$
2. A_1 and utility C jointly perform the Secure_Compare protocol
   If $\text{Secure\_Compare}(a, t) = 1$, then
   (a) A_1 randomly selects r from $\{0, N-1\}$
      - Set $y_1 = N - r$ and $y_2 = a + r \mod N$
      - Send y_1 to $A_2, \ldots, A_n$ and y_2 to C
   (b) Each A_i $(2 \leq i \leq n)$ randomly selects r_i from $\{0, N-1\}$
      - Compute $E(t)^{a_i}$ to get $E(a_i \cdot t)$
      - Set $x_{1i} = N - r_i$ and $s_i = E(a_i \cdot t) \times E(r_i) = E(a_i \cdot t + r_i)$
      - Send s_i to C
   (c) Utility C sets $x_{2i} = D(s_i)$ for $2 \leq i \leq n$
   (d) Each A_i $(2 \leq i \leq n)$ with input (x_{1i}, y_1) and C with input (x_{2i}, y_2) jointly perform the Secure_Divide protocol
      - A_i obtains $\kappa_i = \text{Secure\_Divide}((x_{1i}, y_1), (x_{2i}, y_2))$
      - A_i sets $\delta_i = a_i - \kappa_i$
      - A_i reduces his or her power consumption according to δ_i

Figure 4. f^2_{TPUC} protocol.

Also, we assume that $E(t)$ is initially broadcasted by the utility.

Figure 4 presents the main steps in the f^2_{TPUC} protocol. A_1 is the designated consumer in the participating neighborhood, who is responsible for computing a and distributing $N - r$ to the other consumers and $a + r \mod N$ to the utility. Note that the value of a computed in Step 1 should not include the value a_1 (this is easily achieved via a small modification to the Secure_Sum protocol) and A_1 does not adjust his or her energy consumption. This prevents the disclosure of t to A_1. For instance, if A_1 obtains a δ_1, then A_1 can derive t based on Equation (2). To ensure fairness, A_1 can be randomly selected from among the participating consumers before each execution of the protocol.

The purpose of Step 2(a) is to hide the a value from the utility and the other consumers. Since r is chosen randomly, y_1 and y_2 are randomly distributed in $\{0, N-1\}$. As a result, the other consumers $A_2, \ldots, A_n$ cannot discover a from y_1; similarly, the utility cannot discover a from y_2.

The goal of Step 2(b) is to hide a_i from the utility and t from A_i. Since the encryption scheme is semantically secure, from $E(t)$ and without the private key, the consumers cannot learn anything about t. In addition, because r_i is chosen randomly, the x_{2i} value computed in Step 2(c) does not reveal any information regarding a_i.

The operations performed in Steps 2(b) and 2(c) are based on the additive homomorphic property of the encryption function E. Since $x_{1i} + x_{2i} = a_i \cdot t$ and $y_1 + y_2 = a$, $\kappa_i = \frac{a_i \cdot t}{a}$. Therefore, the protocol correctly returns δ_i for each A_i, except for A_1.

5. Protocol Efficiency and Privacy

This section analyzes the complexity and privacy properties of the protocols.

5.1 Protocol Complexity

Since the Secure_Sum protocol only performs additions and each participant only turns in one input, the protocol is very efficient. The complexity of the Secure_Compare protocol depends on the number of bits needed to represent the maximum value between a and t. Once the number of bits required to represent these numbers is fixed, the complexity of the Secure_Compare protocol is constant. The main operation in the Secure_Max protocol is the comparison of two numbers, so the protocol itself is very efficient. In the case of a neighborhood with 1,000 consumers, if the communication delay is negligible, then the running time of the f^1_{TPUC} protocol is just a few seconds.

According to [1], the computational cost of the Secure_Divide protocol is bounded by $O(\log l)$, where l is the number of bits used to represent the maximum value between $a_i \cdot t$ and a. Because $l = 20$ is generally sufficient in our problem domain, the computational cost of Secure_Divide is constant and very small. If the number of consumers in the neighborhood is small and the utility can execute the Secure_Divide protocol with each consumer concurrently, then the f^2_{TPUC} protocol can also be completed in a few seconds. Based on the above analysis, it is reasonable for the utility to set up a fifteen- or thirty-minute interval between executions of the protocols.

5.2 Protocol Privacy

With regard to the f^1_{TPUC} protocol, a is disclosed to A_1 and a^m is disclosed to all the participating consumers. Since a is aggregated information, the disclosure of a can hardly cause any privacy violations. Although a^m is disclosed, no one can link a^m to a particular consumer. Thus, the disclosure risk of the f^1_{TPUC} protocol is not significant.

The f^2_{TPUC} protocol only discloses a to A_1, so it is more privacy preserving than the f^1_{TPUC} protocol. However, because the Secure_Divide protocol has to be executed between every consumer and the utility, the protocol is less efficient than f^1_{TPUC}. Therefore, depending on whether or not efficiency is more important than privacy, one protocol is more or less applicable than the other protocol in a real-world situation.

6. Conclusions

Intelligent power usage control in the smart grid requires utilities to collect fine-granular energy usage data from individual households. Since this data can be used to infer information about the daily activities of energy consumers, it is important that utility companies and their consumers employ privacy-preserving protocols that facilitate intelligent power usage control while protecting sensitive data about individual consumers.

The two privacy-preserving protocols described in this paper are based on energy consumption adjustment strategies that are commonly employed by utilities. Although the protocols are not as privacy-preserving as the ideal model that engages a trusted third party, they are efficient and limit the amount of information disclosed during their execution. Our future research will focus on refining these protocols and will develop privacy-preserving protocols for other types of energy usage control.

Acknowledgements

The research efforts of the first two authors were supported by the Office of Naval Research under Award No. N000141110256 and by the NSF under Grant No. CNS 1011984. The effort of the third author was supported in part by the Future Renewable Electric Energy Distribution Management Center, an NSF Engineering Research Center, under Grant No. EEC 0812121; and by the Missouri S&T Intelligent Systems Center.

References

[1] M. Atallah, M. Bykova, J. Li, K. Frikken and M. Topkara, Private collaborative forecasting and benchmarking, *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, pp. 103–114, 2004.

[2] A. Ben-David, N. Nisan and B. Pinkas, FairplayMP: A system for secure multi-party computation, *Proceedings of the Fifteenth ACM Conference on Computer and Communications Security*, pp. 257–266, 2008.

[3] M. Blanton, Empirical Evaluation of Secure Two-Party Computation Models, CERIAS Technical Report TR 2005-58, Center for Education and Research in Information Assurance and Security, Purdue University, West Lafayette, Indiana, 2005.

[4] S. Drenker and A. Kader, Nonintrusive monitoring of electric loads, *IEEE Computer Applications in Power*, vol. 12(4), pp. 47–50, 1999.

[5] O. Goldreich, *The Foundations of Cryptography, Volume II: Basic Applications*, Cambridge University Press, Cambridge, United Kingdom, 2004.

[6] O. Goldreich, S. Micali and A. Wigderson, How to play any mental game, *Proceedings of the Nineteenth Annual ACM Symposium on the Theory of Computing*, pp. 218–229, 1987.

[7] S. Goldwasser, S. Micali and C. Rackoff, The knowledge complexity of interactive proof systems, *Proceedings of the Seventeenth Annual ACM Symposium on the Theory of Computing*, pp. 291–304, 1985.

[8] M. Lisovich, D. Mulligan and S. Wicker, Inferring personal information from demand-response systems, *IEEE Security and Privacy*, vol. 8(1), pp. 11–20, 2010.

[9] P. Paillier, Public key cryptosystems based on composite degree residuosity classes, *Proceedings of the Seventeenth International Conference on the Theory and Application of Cryptographic Techniques*, pp. 223–238, 1999.

[10] E. Quinn, Privacy and the New Energy Infrastructure, CEES Working Paper No. 09-001, Center for Energy and Environmental Security, University of Colorado Law School, Boulder, Colorado, 2009.

[11] E. Quinn, Smart Metering and Privacy: Existing Law and Competing Policies, Report for the Colorado Public Utilities Commission, University of Colorado Law School, Boulder, Colorado, 2009.

[12] Smart Grid Interoperability Panel, Guidelines for Smart Grid Security (Introduction, Volumes 1, 2 and 3), NIST IR 7628, National Institute of Standards and Technology, Gaithersburg, Maryland, 2010.

[13] L. Xiong, S. Chitti and L. Liu, Topk queries across multiple private databases, *IEEE Security and Privacy*, pp. 145–154, 2005.

[14] A. Yao, Protocols for secure computations, *Proceedings of the Twenty-Third Annual Symposium on the Foundations of Computer Science*, pp. 160–164, 1982.

[15] A. Yao, How to generate and exchange secrets, *Proceedings of the Twenty-Seventh Annual Symposium on the Foundations of Computer Science*, pp. 162–167, 1986.

Chapter 11

EFFECTS OF TIME DELAYS IN THE ELECTRIC POWER GRID

Hasan Ali and Dipankar Dasgupta

Abstract Communication delays in an electric power grid affect the performance of control systems and can cause power losses. This paper analyzes the causes and effects of communication delays. The analysis employs a simulated power network comprising several generators for which braking resistors with intelligent (fuzzy logic) controllers are used for transient stability control. A balanced 3LG (three-phase-to-ground) fault at different points on the transmission lines is considered. Simulation results show that, although a fuzzy-logic-controlled braking resistor can improve transient stability, the communication delay associated with the online calculation of the total kinetic energy deviation, which is the input parameter for fuzzy control, has an influence on the transient stability performance. The paper also examines the delay that a system can tolerate and the cyber attacks that can cause additional delays.

Keywords: Power grid, communication delays, transient stability control

1. Introduction

In a modern electric power grid, numerous parameters are measured and communicated for purposes of control. In fact, the measurement and communications network of a power system is referred to as a "wide-area measurement system"(WAMS) [12]. Due to the increased speed of communications equipment and the introduction of new devices such as phasor measurement units, some power engineers have proposed that existing wide-area measurement systems be used to implement wide-area controls. Such a wide-area control system (WACS) can be used to control a variety of components, including power system stabilizers, HVDC systems and supplementary controllers for flexible AC transmission system (FACTS) devices.

In a wide-area control system, the time required to transmit data from the measurement location to a control center or data concentrator and the time required to ultimately communicate this data to control devices are collectively

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 139–154, 2012.

referred to as the communication delay or latency [12]. Communication delays can occur at various points in a control system. The introduction of a time delay in a feedback loop has a destabilizing effect and reduces the effectiveness of control system damping. In some cases, system synchronization may be lost [16].

In order to satisfy the performance specifications of wide-area control systems, it is important that delays are taken into account during the controller design. A designed controller should tolerate the specified range of operating conditions and the delay uncertainty [16]. The impact of time delays on controller robustness has largely been ignored in the power systems domain, but the subject has become significant in recent years due to proposals that advocate wide-area power system control.

This paper describes the causes and effects of communication delays in an electric power grid. Also, it examines the amount of delay that a system can tolerate and the cyber attacks that can cause additional delays. The analysis of communication delays is conducted using a simulated power system with generators that employ braking resistors [1] for transient stability control. The switching of braking resistors is implemented using intelligent (fuzzy logic) controllers. The total kinetic energy deviation (TKED) of a generator is used as input to a fuzzy controller for braking resistor switching [2]. Simulations are conducted using the Electro-Magnetic Transients Program (EMTP), a special transient simulation system that can predict the values of variables in an electric power network as functions of time, typically following some disturbance such as the switching of a circuit breaker or a fault [4]. The effectiveness of intelligent braking resistors is demonstrated using a balanced 3LG (three-phase-to-ground) fault at different points on the transmission lines. Various values of communication delays, potentially caused by natural disasters, faults and cyber attacks, are also considered in the transient stability analysis.

2. Communication Delays in the Power Grid

A communication delay in a power grid is defined as the time between the sending of a message from the source device to the receiving of the message at the destination device [14]. It is measured end-to-end between two applications running at the source and destination systems. Because electric power devices do not have communications capabilities, each device is typically attached to an embedded computer system that serves as the communications interface to the network infrastructure.

The electric device and the embedded computer system together form an intelligence electronic device (IED). Figure 1 shows the message processing steps that occur within an IED. In the figure, a message containing device status data is generated and transmitted through four modules in the IED: (i) the analog-to-digital converter transforms the status measurement into digital data; (ii) the CPU processes the measurement data; (iii) the setpoint structure stores the measurement data; and (iv) the network protocol stack formats the

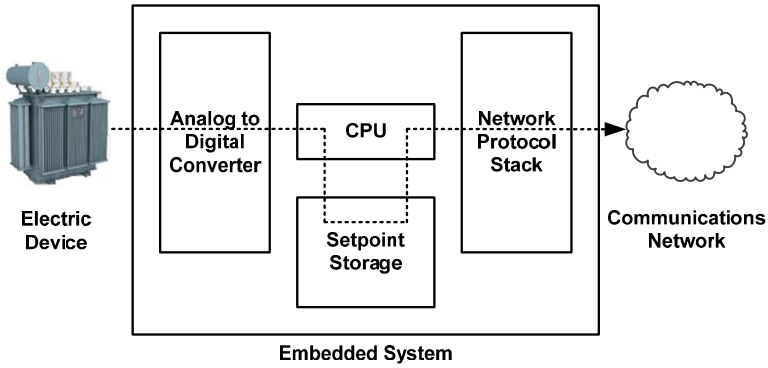


Figure 1. Processing time in an intelligent electronic device.

message and sends it over the network. The time spent by the message within the IED is included in the end-to-end delay.

2.1 Critical Timing Requirements

Timing is critical in power grid communications, more so in the "smart grid." Indeed, this is the most fundamental difference between communications in the power grid and communications in most other networks. Some types of information exchange between electric devices are useful only within a predefined time window. If the communication delay exceeds the time window, the information does not serve its purpose; in the worst case, the delay could damage equipment in the grid.

An example is power device protection, where a circuit breaker must be opened immediately if the voltage or current in a device exceeds a threshold. Rigorous communication delay requirements have been specified for various types of information exchange in power grids (including smart grids). The mandated timing requirements must be met by power grid communications infrastructures.

2.2 Delay Components

The communications infrastructure in a power grid incorporates many networking technologies and has a hierarchical and hybrid composition. Various types of interconnected networks are used to provide communications in different regions of the grid. The delay experienced by a message includes many components as the message travels within each network and through the interfaces between networks. In general, the delay can be categorized in terms of five components [14]:

- **Data Acquisition Delay:** Status measurements, such as voltage, current and temperature, are acquired periodically from electric devices and converted from their original analog formats to digital representations.

The digital information is processed by the attached embedded system, which functions as a low-profile computer, for transmission through the communications networks. The data acquisition delay is the time between the event occurrence (e.g., voltage change) and the actual digital information capture.

- **Packet Processing Delay:** Data is transmitted through a communications network according to the specified network protocols. Various packet headers and trailers are added, inspected, modified and removed along the packet transmission path. Each step in packet processing adds a delay to the total time spent by a packet in the network.

- **Packet Transmission Delay:** Current link layer mechanisms append a data integrity check field to each data frame to detect possible data errors. Every intermediate node on the packet transmission path verifies data correctness after receiving the complete data frame and before forwarding the packet to the next relay node. Each link incurs a transmission delay due to the sending and receiving of a data frame.

- **Medium Access Delay:** Multiple nodes that share the same transmission medium (e.g., wireless spectrum and wireline cable) compete for medium access in order to transmit their data. A node has to wait for its turn to transmit data. Similarly, a packet at a node has to wait until all the other packets scheduled ahead of it have been cleared from the buffer.

- **Event Response Delay:** Some types of IED status messages require actions to be performed. For example, a measured voltage that exceeds the normal value must trigger a circuit breaker command from the control station. The event response delay is the time taken by the intelligent energy and fault management system that resides at the node responsible for the action to actually perform the action.

2.3 Time Delay Calculations

In a wide-area control system, it is assumed that data is transmitted in the form of packets [12]. The packets are formatted blocks of information that are typically arranged in three sections: header, payload and trailer. The information in the header includes the packet length, origin and destination address, packet type and packet number (if a sequence of packets is sent). The payload carries the measurement or control data. The trailer at the end of the packet carries information that enables the receiving device to identify the end of the packet.

The total time delay includes several delays that occur in communications systems [12]. These delays include: (i) serial delay (delay between successive bits); (ii) between packet serial delay (delay between successive packets); (iii) routing delay (time required for data to be sent through a router and then resent to another location); and (iv) propagation delay (time required to transmit data

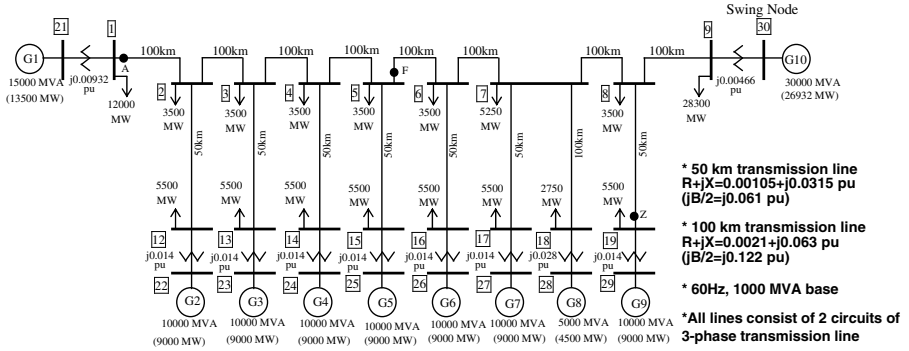


Figure 2. IEEJ West ten-machine model.

over a particular communications medium). The total signal time delay T is given by:

$$T = T_s + T_b + T_p + T_r$$
$$T_s = \frac{P_s}{D_r}$$
$$T_p = \frac{l}{v}$$

where T_s is the serial delay, T_b is the between packet delay, T_p is the propagation delay, T_r is the routing delay, P_s is the size of the packet, D_r is the data rate of the network, l is the length of the communications medium, and v is the speed at which the data is sent though the communications medium (e.g., $0.6c$ to c, where c is the speed of light).

3. Analysis of Communication Delays

The IEEJ West ten-machine model [1] shown in Figure 2 is used to analyze the effects of communication delays. This ten-machine tandem model is a prototype of the Japanese 60 Hz system that presents the long-term oscillation characteristics of a tandem system. The model system incorporates ten generators (G1 to G10). Generator G10 is considered to be the "swing generator." Each line in the figure represents two circuits of a three-phase transmission line. In this work, five braking resistors are installed at the terminal buses of Generators G1, G4-G6 and G10 to stabilize the overall system [2].

Figure 3 shows a braking resistor (BR) with a conductance value of G_{TCSBR} connected via a thyristor switching circuit to one phase of a generator terminal bus. The switching of the braking resistor is accomplished by a fuzzy logic controller. The total kinetic energy deviation (TKED) is used as the input to the fuzzy controller for switching. In our work, TKED is defined as the

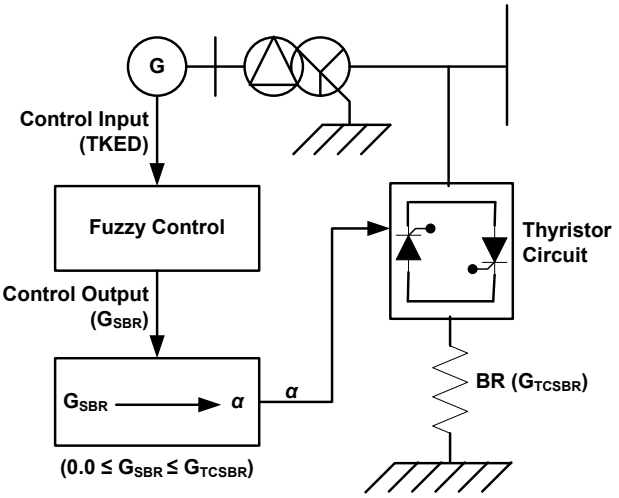


Figure 3. Braking resistor with thyristor switching circuit.

difference between the total kinetic energy (W_{total}) of a generator at a transient state and the total kinetic energy at the steady state. Models of the automatic voltage regulator and governor control system for the IEEJ West ten-machine system were included in our simulation. Interested readers are referred to [1] for details about the generator parameters.

3.1 Closed-Loop Control System

Figure 4 shows a closed-loop control system for braking resistor operation. The communication delay in this system includes the upstream and downstream links. As shown in the figure, the speed equivalent signal of each generator is passed through a filter and an analog-to-digital converter. The resulting digital signals are sent to a central control office, where a global positioning system (GPS) receiver synchronizes the signals using a common timing reference. The synchronized signals are used to compute the TKED, which is sent as input to the fuzzy controller. The signals may be transmitted and received through microwave or optical links.

3.2 Causes of Communication Delays

In the control system in Figure 4, time delays are introduced due to signal transmission via microwave or optical links, analog-to-digital conversion, online TKED computation and time synchronization of GPS signals. These communication delays adversely affect the opening and closing of circuit breakers following a fault in the electric grid. Note that communication delays may also result from attacks on the information infrastructure; these delays are discussed in Section 6.

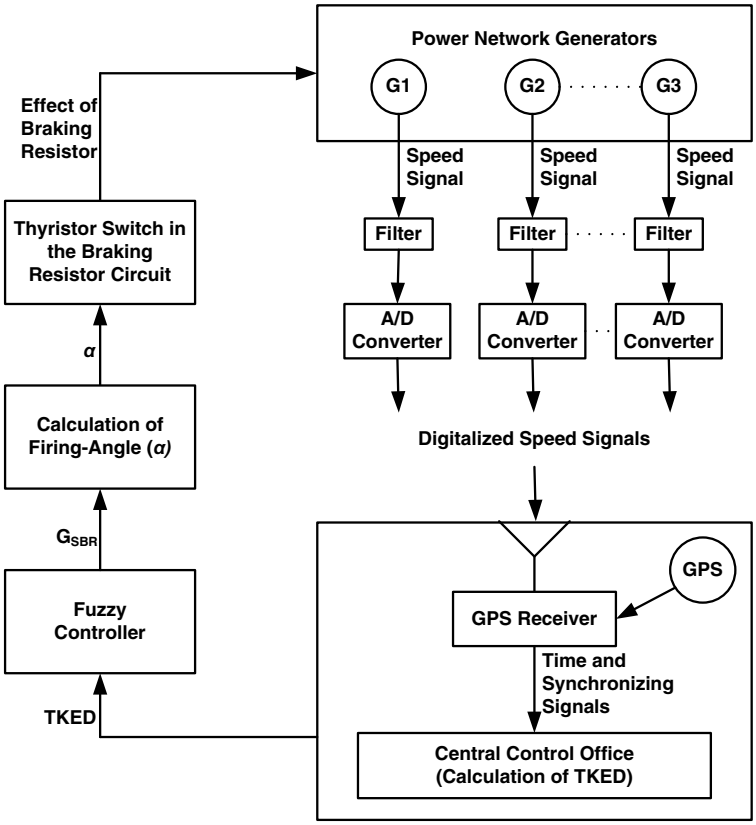


Figure 4. Closed-loop control system with GPS functionality.

3.3 Delay Range

Communication delays typically range from several microseconds to a few hundred milliseconds [3, 13, 15, 17]. In a distributed system such as a protective relay system, the time delay or latency is usually less than 10 ms [16]. Unlike the small time delays encountered in local control, the delays in wide-area power systems can range from tens to several hundred milliseconds or more. In the Bonneville Power Administration system, the latency of fiber optic digital communications is approximately 38 ms for one way, while the latency when using modems via microwave is more than 80 ms [16]. Communications systems that use satellites may have even longer delays.

The delay of a signal feedback in a wide-area power system is usually in the order of 100 ms [16]. If routing delays are included and if a large number of signals are to be routed, then there is the potential to experience long delays and considerable variability (or uncertainty) in these delays. According to some reports (see, e.g., [8, 10]), communication delays of 150 to 200 ms

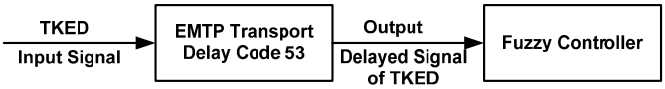


Figure 5. Applying communication delays to a controller.

should be considered when designing some transient stability control systems. In this work, simulations are conducted with communication delays ranging from 20 μsec to 500 msec.

3.4 Implementation of Communication Delays

The simulations were conducted using the Electro-Magnetic Transients Program (EMTP). During the simulations, various values of communication delays were applied to the fuzzy controller input signal through the EMTP Transient Analysis of Control Systems (TACS) Code 53 (i.e., EMTP transport delay code). The procedure is illustrated in the block diagram in Figure 5. According to the EMTP transport delay code 53, at any time t, for a value of total delay t_d (sec), the following equation holds [4]:

$$Output(t) = Input(t - t_d)$$

3.5 Delay Realities

The networking infrastructures currently in use were not designed with communication delay performance as a priority and, therefore, they may not always be able to meet the strict delay requirements of power system communications. Preliminary results of experiments on communication delays in substation networks are reported in [7]. The results demonstrate that, in many communications scenarios, the packet delays experienced in typical substation networks exceed the maximum allowed for the most time critical messages. Also, while communication delays within a single Ethernet segment are below 2 ms, the delays increase significantly in wireless and multihop networks.

Communication delays in substations have also been investigated using simulations [11]. The simulation results show that 10/100 Mbps Ethernet networks can provide satisfactory delay performance for communications in a substation. The delay measurements for the simulated network settings are less than 1 ms in most cases, which are consistent with the experimental results on Ethernet networks reported in [7]. Also, it has been observed that communication delays increase with the distance between communicating devices and, therefore, delays in large Ethernet networks may need further investigation.

Single-hop WiFi networks cannot be used to transmit system protection messages, but these networks meet the delay requirements of all other messages (e.g., system monitoring and control, operation and maintenance, text files, images and videos). ZigBee networks and multihop networks with wireless access, however, can only be used to transmit data that is not time sensitive.

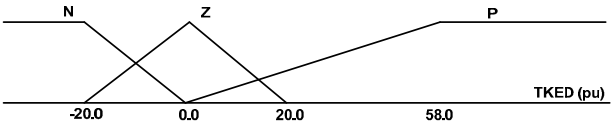


Figure 6. Membership functions for controller input $TKED$.

In particular, these networks cannot be used to transport system protection, monitoring and control messages. The delay performance becomes even worse when networks experience heavy background traffic loads or when complex multihop networks are used. Hence, the design of short-delay networks is a critical problem to support effective energy management functions, especially in the smart grid.

4. Fuzzy Logic Controller Design

Fuzzy logic extends two-valued Boolean logic by permitting truth values in the continuous interval [0,1] where 0 is completely false, 1 is completely true, and the values in between 0 and 1 express degrees of truth. This extension is especially useful for solving problems that involve subjective uncertainty or imprecision.

Fuzzy logic control is a process control paradigm that is based on fuzzy logic. It typically employs a series of IF-THEN rules, whose conditions and actions are expressed as fuzzy variables. This section describes the design of the fuzzy logic controllers used for switching the braking resistors.

4.1 Fuzzification

Each fuzzy logic controller has as input the $TKED$ value of the associated generator and produces an output G_{SBR}, the braking resistor conductance, where $G_{SBR} \in [0, G_{TCSBR}]$. Triangular fuzzy membership functions are chosen for $TKED$ as shown in Figure 6. The membership functions specify the fuzzy linguistic variables $Negative$, $Zero$ and $Positive$, which are denoted as N, Z and P, respectively. The precise shapes of the membership functions for $TKED$ are determined by trial and error in order to obtain good performance.

The triangular membership functions μ_A used to determine the fuzzy membership values of input variable values $TKED$ ranging from -20 pu to 58 pu are given by [5]:

$$\mu_A(TKED) = \frac{1}{b}\left[b - 2 \times \left|TKED - a\right|\right] \tag{1}$$

where $\mu_A(TKED)$ is the membership value corresponding to a controller input value $TKED$, b is the width of the membership function and a is the coordinate of the point at which the membership value is one.

Table 1. Fuzzy rule table.

TKED [pu]	G_{SBR} [pu]				
	BR1	**BR4**	**BR5**	**BR6**	**BR10**
N	0.0	0.0	0.0	0.0	0.0
Z	0.0	0.0	0.0	0.0	0.0
P	15.0	7.0	4.0	4.0	4.0

4.2 Control Rules

The fuzzy control strategy is very simple because it incorporates only three IF-THEN control rules for each controller. Table 1 shows the control rules, where the numerical values of G_{SBR} correspond to the fuzzy controller outputs. Note that the control rules were developed by trial and error based on practical system operations. For example, a braking resistor (BR) can consume active power during acceleration (P: *Positive*) but cannot generate or consume active power during deceleration (N: *Negative*). Also, when the system is at steady state (Z: *Zero*), the braking resistor does not need to consume active power. Thus, the fuzzy rule table only has P (*Positive*) values.

4.3 Fuzzy Inference

Mamdani's inference mechanism [5] is employed by the fuzzy logic controller. According to this mechanism, the degree of satisfaction W_i of a fuzzy rule i is given by:

$$W_i = \mu_A(TKED)$$

where $\mu_A(TKED)$ is the membership value as defined by Equation (1).

4.4 Defuzzification

A defuzzification method is required to determine the crisp (precise) output value of the controller, i.e., the conductance value G_{SBR} of the braking resistor. The center-of-area method [5], a simple and popular defuzzification method, is used in this work. According to this method, the controller output value G_{SBR} is given by:

$$G_{SBR} = \frac{\sum W_i C_i}{\sum W_i}$$

where C_i is the value of G_{SBR} in the fuzzy rule table (Table 1).

4.5 Thyristor Firing Angle Computation

The firing angle α of the thyristor switch is calculated from the output of the fuzzy controller, i.e., from the conductance value of the braking resistor $G_{SBR} \in [0, G_{TCSBR}]$. The conductance value G_{SBR} is related to the power dissipated in the braking resistor.

At any step in the simulation, P_{SBR}, the average power of a system braking resistor (SBR) with a conductance of G_{SBR}, is equal to P_{TCSBR}, the average power of a thyristor controlled system braking resistor (TCSBR) with a conductance of G_{TCSBR}. Thus, the firing angle α can be computed using the equation:

$$P_{TCSBR} = P_{SBR}$$

or

$$\frac{V_g^2 G_{TCSBR}}{\pi}(\pi - \alpha + 0.5\sin(2\alpha)) = V_g^2 G_{SBR}$$

where V_g is the root-mean-square value of the generator terminal bus voltage.

5. Simulation of Communication Delays

Communication delays can affect the control logic and, consequently, the performance of the overall system. Therefore, it is important to consider communication delays in a study of a power network.

We conducted simulations using balanced (3LG) faults at Points A, F and Z on the transmission lines. In all the test cases, the simulated fault occurred at 0.1 sec, the circuit breakers on the faulty lines were opened at 0.17 sec, and the circuit breakers were closed at 1.003 sec. It was assumed that the circuit breaker cleared the line when the current through it crossed the zero level. The time step and simulation time were chosen to be 0.00005 sec and 20 sec, respectively.

In order to understand the effects of communication delays, we conducted several experiments that ignored communication delays. The transient stability of the system was evaluated using a stability index W_c (lower W_c value indicates better performance). The stability index (sec) is given by:

$$W_c = \int_0^T \left| \frac{d}{dt} W_{total} \right| dt / system\, base\, power$$

where T is the simulation time of 20 sec and W_{total} is the total kinetic energy (Joules) given by:

$$W_{total} = \sum_{i=1}^{N} W_i$$

Table 2. Values of W_c with communication delays.

Fault Point	Communication Delay	W_c (sec) with BR	W_c (sec) without BR
A	20 μsec	26.353	238.917
	200 μsec	29.331	
	2 msec	30.133	
	20 msec	32.786	
	200 msec	39.401	
	500 msec	47.855	
F	20 μsec	30.527	72.573
	200 μsec	33.613	
	2 msec	35.565	
	20 msec	36.116	
	200 msec	37.657	
	500 msec	38.282	
Z	20 μsec	24.267	70.135
	200 μsec	26.175	
	2 msec	31.063	
	20 msec	33.127	
	200 msec	40.480	
	500 msec	40.536	

in which the kinetic energy of the i-th generator W_i (Joules) is given by:

$$W_i = \frac{1}{2} J_i \omega_{mi}^2$$

Note that N is the total number of generators. Also, the moment of inertia J_i (kg·m^2) is given by:

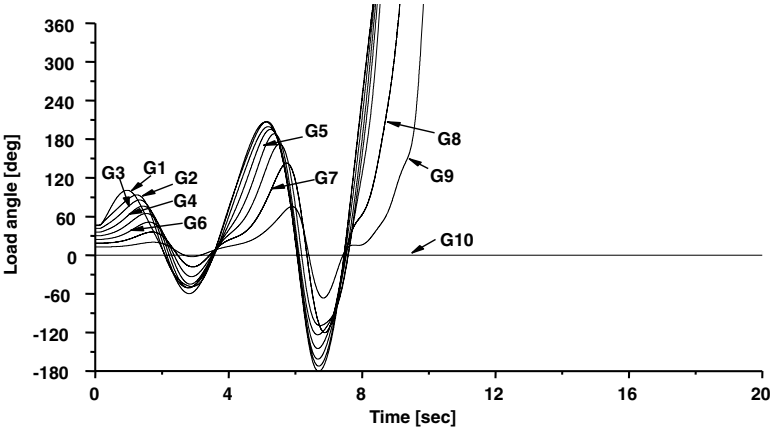
$$J_i = \frac{H \times MVA\ rating}{5.48 \times 10^{-9} N_S^2}$$

where N_S and H are the synchronous angular speed (rpm) and inertia constant, respectively, and

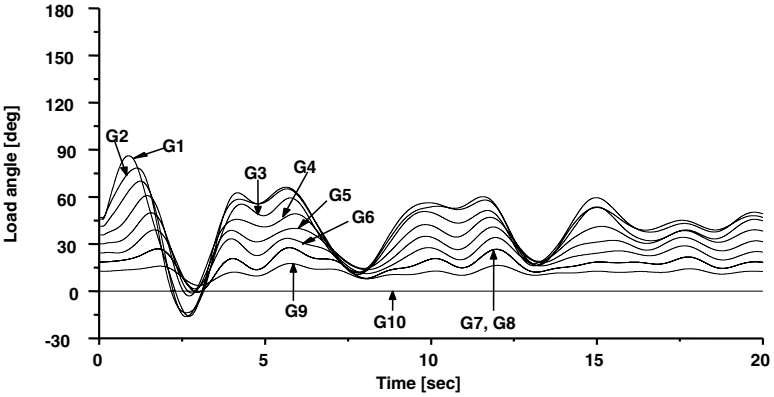
$$\omega_{mi} = \frac{2\pi N_R}{60}$$

is the rotor angular velocity (mechanical rad/sec) and N_R is the rotor speed (rpm).

Extensive simulations were conducted to perform the transient stability analysis. Table 2 shows the W_c values for 3LG faults at Points A, F and Z for various communication delays. The results demonstrate that the fuzzy-controlled braking resistors are effective at improving the transient stability. Also, the W_c values corresponding to different communication delays are different at different points. This indicates that the communication delay associated with the



(a) Without braking resistor.



(b) With fuzzy-logic-controlled braking resistor.

Figure 7. Load angle responses without communication delays.

online calculation of the fuzzy controller input has a small negative impact on the transient stability.

Figures 7(a) and 7(b) show the load angle responses in the case of a 3LG fault at Point A without and with a fuzzy-controlled braking resistor, respectively. Communication delays were not considered in this case. The responses demonstrate that the system is transiently stable when the fuzzy-controlled braking resistor is used.

Figure 8 shows the load angle responses with a fuzzy-controlled braking resistor for a 3LG fault at Point A and a communication delay of 500 msec. The transient stability in this case is worse than that shown in Figure 7(b), where there was no communication delay. This result shows that communication delays do, indeed, affect the transient stability performance.

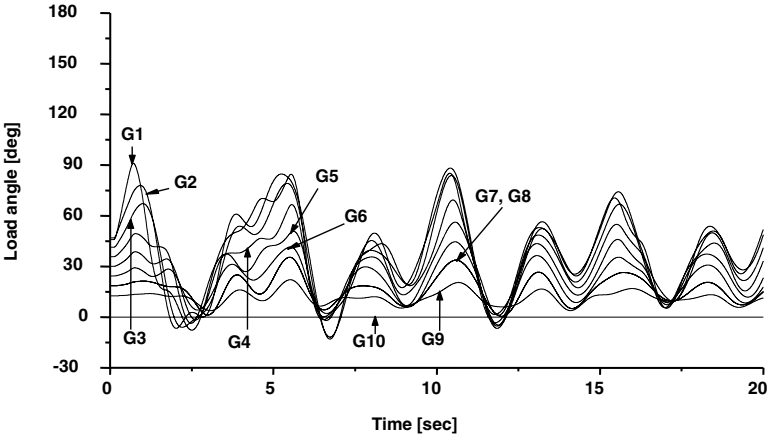


Figure 8. Load angle responses for a communication delay of 500 msec.

It is important to note that each system has a specific delay tolerance for it to function properly. Some systems can tolerate delays of 200 msec while others can function with delays of up to 300 msec. However, in the case of the power system model considered in this work, if the delay exceeds 500 msec, then the system performance deteriorates and the system becomes marginally stable. Therefore, the maximum allowable delay for the system is 500 msec.

6. Attacks Causing Communication Delays

The backbone of a power grid (and especially a smart grid) is the information infrastructure that is used for communications by the various grid components. The power industry uses different types of media (leased lines, wide-area networks, Internet, radio and microwave) to transmit data and signals between control centers and remote locations. The digital signals must be highly synchronized and time-aligned using accurate GPS clocks. However, some of the communications links are highly vulnerable to denial-of-service and man-in-the-middle attacks. Also, an attacker with unauthorized access could inject false signals to disrupt the supervisory control and data acquisition system (SCADA), resulting in power system instability.

An attacker can also use distributed denial-of-service attacks to delay, block or jam the flow of command and control messages in communications networks. Sophisticated malware such as Stuxnet can disrupt communications and synchronization, potentially resulting in massive instabilities in the power grid. These threats will be even more serious in the smart grid where communication delays must be small, and where additional delays are introduced by security measures such as encryption and authentication [6, 9].

7. Conclusions

Electric power grids require an extensive information and communications infrastructure to support the efficient and safe generation, transmission and distribution of electricity. However, the associated communication delays can affect the performance of control systems, causing power losses and possibly equipment damage. The simulation results using the IEEJ West ten-machine model demonstrate that fuzzy-logic-controlled braking resistors are highly effective at improving transient stability. But it is important to note that the delay associated with online calculations of the total kinetic energy deviation (fuzzy controller inputs) can have a negative impact on transient stability performance.

Our future research will investigate the negative effects of delays resulting from faults, failures and cyber attacks. It will also examine defensive strategies involving the use of monitoring, traffic analysis and response tools.

References

[1] H. Ali, T. Murata and J. Tamura, The effect of temperature rise of fuzzy-logic-controlled braking resistors on transient stability, *IEEE Transactions on Power Systems*, vol. 19(2), pp. 1085–1095, 2004.

[2] H. Ali, T. Murata and J. Tamura, Effect of coordination of optimal reclosing and fuzzy-controlled braking resistors on transient stability during unsuccessful reclosing, *IEEE Transactions on Power Systems*, vol. 21(3), pp. 1321–1330, 2006.

[3] B. Chaudhuri, R. Majumder and B. Pal, Wide-area measurement-based stabilizing control of a power system considering signal transmission delays, *IEEE Transactions on Power Systems*, vol. 19(4), pp. 1971–1979, 2004.

[4] H. Dommel, *EMTP Theory Book*, Microtran Power System Analysis Corporation, Vancouver, Canada, 1992.

[5] D. Driankov, H. Hellendoorn and M. Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, New York, 1993.

[6] G. Ericsson, Cyber security and power system communication – Essential parts of a smart grid infrastructure, *IEEE Transactions on Power Delivery*, vol. 25(3), pp. 1501–1507, 2010.

[7] M. Khanna, Communication challenges for the FREEDM System, M.S. Thesis, Department of Computer Engineering, North Carolina State University, Raleigh, North Carolina, 2009.

[8] M. Koaizawa, M. Nakane, K. Omata and Y. Kokai, Actual operating experience of on-line transient stability control systems (TSC systems), *Proceedings of the IEEE Power Engineering Society Winter Meeting*, vol. 1, pp. 84–89, 2000.

[9] National SCADA Test Bed, Study of Security Attributes of Smart Grid Systems – Current Cyber Security Issues, INL/EXT-09-15500, Idaho National Laboratory, Idaho Falls, Idaho, 2009.

[10] H. Ota, Y. Kitayama, H. Ito, N. Fukushima, K. Omata, K. Morita and Y. Kokai, Development of transient stability control system (TSC system) based on on-line stability calculation, *IEEE Transactions on Power Systems*, vol. 11(3), pp. 1463–1472, 1996.

[11] T. Sidhu and Y. Yin, Modeling and simulation for performance evaluation of IEC 61850 based substation communication systems, *IEEE Transactions on Power Delivery*, vol. 22(3), pp. 1482–1489, 2007.

[12] J. Stahlhut, T. Browne, G. Heydt and V. Vittal, Latency viewed as a stochastic process and its impact on wide-area power system control signals, *IEEE Transactions on Power Systems*, vol. 23(1), pp. 84–91, 2008.

[13] C. Taylor, D. Erickson, K. Martin, R. Wilson and V. Venkatasubramanian, WACS – Wide-area stability and voltage control system: R&D and online demonstration, *Proceedings of the IEEE*, vol. 93(5), pp. 892–906, 2005.

[14] W. Wang, Y. Xu and M. Khanna, A survey of the communication architectures in the smart grid, *Computer Networks*, vol. 55, pp. 3604–3629, 2011.

[15] R. Wilson, An investigation of time transfer accuracies over a utility microwave communications channel, *IEEE Transactions on Power Delivery*, vol. 8(3), pp. 993–999, 1993.

[16] H. Wu, K. Tsakalis and G. Heydt, Evaluation of time delay effects to wide-area power system stabilizer design, *IEEE Transactions on Power Systems*, vol. 19(4), pp. 1935–1941, 2004.

[17] W. Yao, L. Jiang, Q. Wu, J. Wen and S. Cheng, Delay-dependent stability analysis of a power system with a wide-area damping controller embedded, *IEEE Transactions on Power Systems*, vol. 26(1), pp. 233–240, 2011.

Chapter 12

MEASURING NAME SYSTEM HEALTH

Emiliano Casalicchio, Marco Caselli, Alessio Coletta,
Salvatore Di Blasi, and Igor Nai Fovino

Abstract Modern critical infrastructure assets are exposed to security threats arising from their use of IP networks and the Domain Name System (DNS). This paper focuses on the health of DNS. Indeed, due to the increased reliance on the Internet, the degradation of DNS could have significant consequences for the critical infrastructure. This paper describes the Measuring Naming System (MeNSa), a framework designed to provide a formal methodology, metrics and tools for evaluating DNS health. Additionally, it proposes a process for aggregating health and security metrics to provide potential threat indicators. Results from a scenario-based experiment demonstrate the utility of the framework and aggregation metrics.

Keywords: Domain Name System, security, aggregation metrics

1. Introduction

Critical infrastructure assets such as electric power grids, gas pipelines, and telecommunications and banking systems are increasingly reliant on information and communications technologies. Information and communication technologies provide opportunities to enhance and optimize services and efficiently manage remote installations. As consequence, however, the information and communications infrastructures that enable these services have become vital to the proper operation of critical infrastructure assets.

This paper focuses on the Domain Name System (DNS) infrastructure. DNS is a hierarchical naming system that "maps" Internet domain names to corresponding IP addresses. Often viewed as a phone book, the operation of DNS is essential to the proper functioning of the Internet. Without DNS, it would be practically impossible for users to navigate the Internet or use web service applications. Due to the growing interconnectivity of critical infrastructure assets, a DNS fault under certain conditions could have serious national and international implications [14].

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 155–169, 2012.

DNS security concerns and the potential impact were discussed during the Internet Corporation for Assigned Names and Numbers (ICANN) symposia in 2009 and 2010 [17, 18]. From these two symposia emerged the concept of DNS health as a means for expressing the status of DNS.

This paper presents results from the Measuring Naming System (MeNSa) effort [12]. The primary goal of the project is to design a formal methodology, metrics and tools for evaluating DNS health. The paper presents the architecture of the framework [6], sample metrics [7] and the operation schema [5]. Additionally, it describes a process for aggregating health and security metrics via structured indices.

2. Domain Name System

This section provides an overview of DNS. Additionally, it discusses vulnerabilities and the associated impact on information and communications technology infrastructures.

2.1 DNS Overview

The DNS infrastructure is composed of geographical and logical entities that are organized in a hierarchical fashion. The topmost level of the hierarchy is the root domain, while the next subordinate level consists of top-level domains (TLDs). Each TLD, in turn, can have many sub-domains, called second-level or enterprise-level domains. Entities associated with the root domain are called root operators. Registries are the organizations that manage name servers related to a TLD. To facilitate the administration process, DNS defines the concept of a zone – a portion of the domain name space for which administrative responsibility is delegated.

A DNS query to resolve an Internet domain name originates from a client component to either an authoritative name server or a caching name server. Note that this process can be iterative or recursive. A response is generated that provides the IP address corresponding to the Internet domain name. A zone transfer represents an operation where a secondary name server refreshes its records with the primary name servers. This process enables a secondary name server to maintain synchronization with the primary name server. DNS dynamic services provide the ability to dynamically add and/or delete a subset of the resource records for an existing domain, to delete an entire domain, or to create a new domain. DNS administrative services also include tasks performed by the responsible entity to provide an appropriate level of service and to ensure security.

2.2 DNS Threats

DNS was designed in the 1980s with little concern for security. Because DNS functionality has, for the most part, remained unchanged, several intrinsic

vulnerabilities exist. DNS threats can be broadly classified into three main categories [23]: (i) data corruption; (ii) denial of service (DoS); and (iii) privacy.

Data corruption is defined as the unauthorized modification of DNS data and includes repository corruption and system corruption. Repository corruption is the debasement of databases containing authoritative data necessary for DNS operations (e.g., resource records and zone files). System corruption is the alteration of the authenticity of DNS responses. Note that weaknesses in the design of the DNS protocol are often exploited in data corruption attacks. Examples include cache poisoning, route injection and man-in-the-middle attacks. The well-known Kaminsky attack [19] is a concrete example of this class of attacks.

A DoS attack renders a service unavailable to legitimate users. These attacks usually impact a specific service (e.g., targeting assets that rely on the proper functioning of the DNS) or create wide-ranging outages (e.g., degrading general Internet functionality).

A privacy threat relates to the loss or theft of personal information. One example is reading a DNS cache to discern an individual's browsing activities. The consideration of privacy threats is beyond the scope of this paper. However, we intend to consider privacy issues in our future work related to DNS health.

2.3 DNS Incidents

The first security flaws in the DNS protocol were identified in the early 1990s when Bellovin [4] and Vixie [25] discovered how to spoof name-based authentication systems using cache contamination attacks. The security extension DNSSEC was proposed in 1997 to address the identified vulnerabilities [10, 11]. Further cache poisoning vulnerabilities discovered by Kaminsky led to the development of additional specifications, namely RFC 4033 [1], RFC 4034 [2] and RFC 4035 [3].

Two major attacks have been reported on DNS root servers. The first attack, which occurred in 2002 and lasted approximately one hour, simultaneously targeted all thirteen DNS root servers [26]. The performance and availability of nine servers were degraded during the attack; in response, the Anycast protocol was implemented in eleven root servers. The second global attack occurred in 2007 [15]. This attack was larger in scale, however, only the two root servers that had not adopted the Anycast solution were impacted.

Root servers are not the only DNS components that are vulnerable. Several DNS hijacking attacks that targeted domain name registrars have been reported. In June 2008, for example, the ICANN website was the victim of a defacement attack resulting from the compromise of its name registrar [16]. Another attack compromised a large e-bill payment site that redirected visitors to an alternate website and installed malicious code on their machines [20]. In 2009, the New Zealand version of Microsoft's MSN website was compromised after attackers penetrated the country's primary domain name registrar [9]. Similarly, in 2009, a domain name registrar in Puerto Rico was compromised, resulting in the redirection of local websites for major companies such as

Google, Microsoft and Yahoo [22]. Also in 2009, malicious entities used cache poisoning to redirect the login page for a major Brazilian bank to a fraudulent website that stole user credentials [13].

A recent study by the Global Cyber Security Center detailed how a DNS attack could impact operations in a smart grid [14]. Indeed, the increasing dependency of critical infrastructure assets on information and communications technologies warrants security solutions that ensure that DNS is adequately protected.

2.4 DNS Health and Security

The security, stability and resiliency of DNS have received significant attention over the past few years. Following the 2009 and 2010 DNS symposia [17, 18], ICANN specified the following indicators for DNS health:

- **Availability:** The ability of DNS to be operational and accessible when required.
- **Coherency:** The ability of DNS to accurately resolve name queries; this is one of the core principles of DNS. For example, if the IP address `192.0.2.1` is resolved to `www.foo.example.com`, then the coherency principle implies that the name `www.foo.example.com` should resolve to the IP address `192.0.2.1`.
- **Integrity:** The ability of DNS to guard against improper data modification or destruction; this includes ensuring information non-repudiation and authenticity.
- **Resiliency:** The ability of DNS to effectively respond and recover to a known, desired and safe state in the event of a disturbance.
- **Security:** The ability of DNS to limit or protect itself from malicious activities (e.g., unauthorized system access, fraudulent representation of identity and interception of communications).
- **Speed:** The performance of DNS with respect to response time and throughput. Note that, in addition to queries, speed applies to maintenance, administration and management operations.
- **Stability:** The ability of DNS to function in a reliable and predictable manner (e.g., protocols and standards). Stability is important because it facilitates universal acceptance and usage.
- **Vulnerability:** The likelihood that a DNS weakness can be exploited by one or more threats.

Several studies have examined DNS traffic measurement techniques and performance metrics [8, 21, 24]. However, hardly any research has examined DNS

health in relation to the prescribed indicators. This paper focuses on the security, resiliency and vulnerability indicators for deriving DNS health metrics associated with our MeNSa framework.

3. MeNSa Framework

The 2009 and 2010 ICANN DNS symposia that introduced the concept of DNS health, also identified the following requirements:

- The need for viable indicators of DNS health for different DNS actors (i.e., root server operators, non-root authoritative name server operators, recursive caches, open DNS resolvers and end users).
- The need to understand and refine proper methods and techniques for measuring DNS health indicators.
- The need to refine and improve existing metrics for availability, coherency, integrity, resiliency, security, speed, stability and vulnerability.
- The need for metric threshold levels that identify when DNS health has degraded below acceptable standards.

Despite the specification of these requirements, the realization of DNS health metrics is still at a primitive stage. This section describes the main components of the MeNSa framework for deriving DNS health metrics. Interested readers are referred to [5–7] for additional details about MeNSa.

3.1 Framework Components

Figure 1 shows the primary components of the MeNSa framework along with their functional relationships. The DNS reference model specifies the attributes that must be measured in order to discern DNS health levels. Note that the point of view (PoV) is an inherent part of the DNS reference model that specifies DNS health from a local perspective for components and actors. A set of use cases provide detailed scenarios that outline the functional interactions between DNS components and actors. Measurement techniques and tools specify methods for obtaining the information necessary to compute the metrics. Metrics are derived that quantify DNS health based on inputs from the other primary components.

3.2 Reference Architecture

Figure 2 presents a graphical display of the reference DNS architecture. The user application (e.g., Internet browser) is the actor that generates DNS queries. The application service provider is the actor that provides distributed services and applications, primarily via web service technologies. The name server resolves queries for a specific zone and can function as a master or slave. The resolver is a name server, often owned and managed by an Internet service provider (ISP), that receives DNS queries and either resolves the queries or

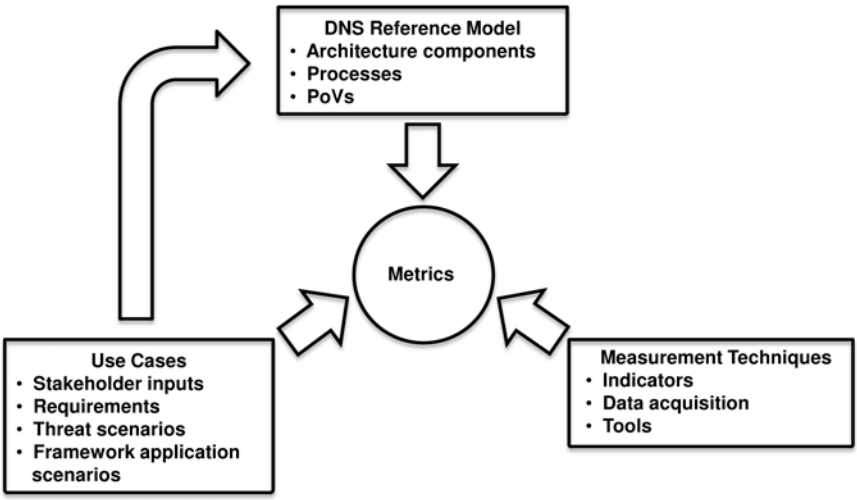


Figure 1. MeNSA framework.

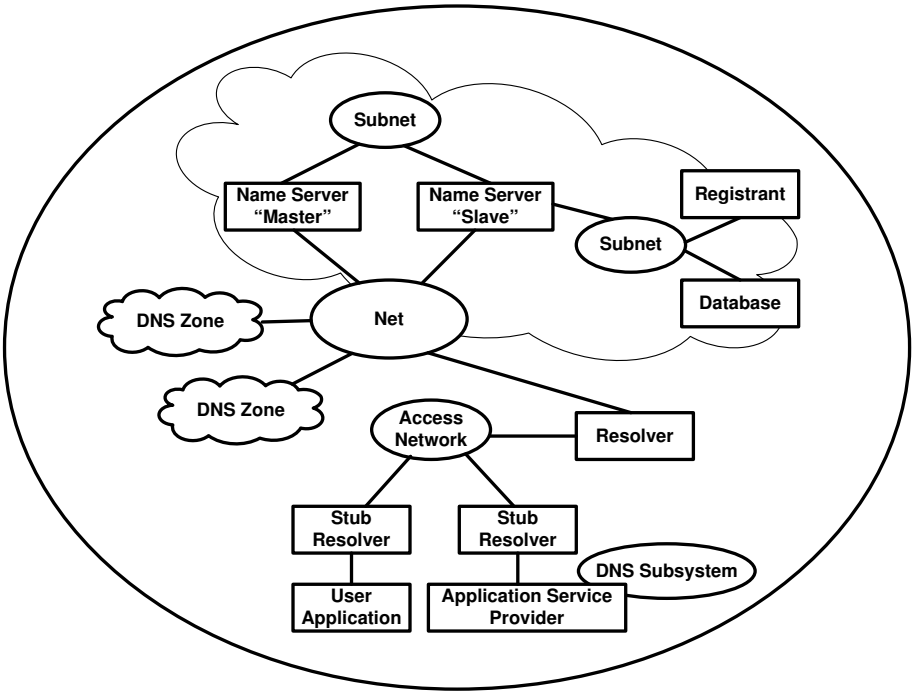


Figure 2. Reference architecture.

forwards them to the next server in the DNS hierarchy. The stub resolver is the operating system component that receives DNS requests from applications and sends them to the resolver. The net and subnet components represent the various network interconnections (e.g., LAN and Internet communication channels). The registrant represents the administrator of a zone. Databases store DNS information for each respective zone. The DNS zone is a specific DNS domain that is managed by a single administrative entity. Finally, the DNS subsystem represents an autonomous naming system that is isolated from the global DNS.

3.3 Point of View

The MeNSa framework is intended to provide DNS health awareness for end users, application service providers and operators (e.g., resolvers, name servers and registrars). Depending on their role and access to various components, each actor may have one or more views of DNS. Note that the perception of health is limited in scope by processes and components that each actor can observe and control.

The PoV concept helps categorize components that a specific DNS actor can observe and measure. Additionally, PoVs help identify the information that is required from other actors to properly assess DNS health. The six PoVs incorporated in the MeNSa framework are: (i) end user PoV; (ii) application service provider PoV; (iii) resolver PoV; (iv) name server PoV; (v) zone PoV; and (vi) global PoV.

Of particular interest in this work is the end user PoV, which represents the perspective from which each user can evaluate DNS. The components associated with the end user PoV are the user application, stub resolver and net. The specific operation of interest is the DNS lookup process.

3.4 Metrics

The proposed metrics are intended to evaluate DNS health based on vulnerability, security and resiliency. Table 1 provides example categories and metrics associated with the MeNSa framework. The vulnerability metrics are associated with repository corruption, system corruption and denial of service. Indicators for repository corruption include data staleness, zone drift/zone thrash and data coherence. System corruption indicators include zone transfer failure, DNS spoofing and cache poisoning. Denial of service indicators include DNS request variation, bandwidth consumption and traffic variation. Metrics for resiliency include indicators for mean time to discovery, mean time between failures and operational availability. Finally, security metrics are associated with indicators for attack surface, attack depth, attack escalation speed and annual loss expectancy. Interested readers are referred to [7] for a comprehensive list of derived metrics used in the MeNSa effort.

Table 1. DNS health and security metrics.

Indicator	Metric
Data Staleness	Percent of resource records that differ across authoritative servers
Zone Drift/Zone Thrash	Probability of incurring zone drift and zone thrash
Data Coherence	Percent of responses that differ between queries to the parent zone and authoritative server
Zone Transfer Failure	Number of failed zone transfer operations
DNS Spoofing	Probability of being spoofed
Cache Poisoning	Percent of content that differs between cache and authoritative data
DNS Request Variation	Variance of the number of requests per second
Bandwidth Consumption	Percent of available bandwidth
Traffic Variation	Variance of the incoming DNS traffic rate
Mean Time to Discovery	Average response time
Mean Time between Failures	Average time between invalid responses
Operational Availability	Percent of time executing at the expected service level
Attack Surface	Percent of nodes vulnerable to a certain type of attack
Attack Depth	Percent of nodes impacted by an attack
Attack Escalation Speed	Time required to affect a specified number of nodes
Annual Loss Expectancy	Financial loss as a result of incidents in one year

3.5 Framework Application

This section describes the main phases of the MeNSa framework and how the framework can be used in an operational environment. The application of the framework is organized into three macro phases: (i) preliminary diagnosis; (ii) service level objectives (SLOs) and scenario definition; and (iii) detailed diagnosis and measurement.

In the preliminary diagnosis phase, an initial evaluation of DNS health is conducted based on a subset of the metrics associated with the respective PoV. In the SLOs and scenario definition phase, one or more threat scenarios are derived given the PoV and representative indices. The detailed diagnosis and measurement phase assesses the perceived health level, achievable SLOs, causes of SLO violations and improvement actions.

The detailed diagnosis and measurement phase is further organized into three stages: (i) metric selection; (ii) measurement; and (iii) aggregation. The selection of metrics is an off-line process. The MeNSa framework enables users to predefine a set of validated metrics for each perceived threat scenario and PoV. The measurement stage involves data collection and the computation of the selected metrics. Note that we use a "bottom up" measurement model [5, 6] that first acquires information from other PoVs. Certain indices (e.g., network reachability and traffic load) help discern if a measurement can be effected by critical states of the infrastructure.

The aggregation stage combines the results from the measurement stage to provide aggregated indices that summarize DNS health as perceived by the

PoV. The indicators determine achievable SLOs, causes of health degradation and possible solutions. In the MeNSa framework, data aggregation is accomplished according to the following definitions:

- $M = \{m_1, \ldots, m_M\}$ is the set of metrics used to evaluate DNS health and security.
- D_i is the domain of the i-th metric.
- $v_{i,j} \in D_i$ defines values for the metric m_i. Note that $j = 1, \ldots, n$ where n is the number of computed values.
- $q_i \colon D_i \rightarrow [0, 1]$ is a "quality mapping" for metrics m_i with q_i transforming the measured values $v_{i,j}$ into a dimensionless quality value $q_{i,j} = q_i(v_{i,j})$. Note that $q_{i,j} = 1$ is the highest quality value and $q_{i,j} = 0$ is the lowest quality value.
- $\{w_k\}$ is a set of "weight vectors," where $w_k = (w_{k,1} \ldots w_{k,M})$ is a vector of weights such that $w_{k,i} \in [0, 1]$ and $\sum_{i=1}^{M} w_{k,i} = 1$. Each vector w_k defines the aggregation of the M metrics corresponding to the k-th result.

Given the above definitions, the aggregation process can be specified as:

1. Choose a set of metrics to be aggregated and calculate n $v_{i,j}$ values.
2. Define a quality mapping q_i for each metric and transform the measured values into quality values $q_{i,j} = q_i(v_{i,j})$.
3. Aggregate the quality metrics by averaging the quality values using a weights vector v_k.

4. Experimental Evaluation

This section evaluates the utility of the MeNSa framework and the application of the associated metrics. A scenario-based experiment is used to demonstrate how a subset of defined metrics can be computed and aggregated for the end user PoV.

4.1 Measurements and Metrics

The experimental testbed consisted of a Windows machine running Firefox 8.0 and connected to the Internet through the Italian ISP Fastweb (7 Mbps nominal). DNS queries were sent to Fastweb's recursive resolvers.

Data was collected during ten web browsing sessions ranging in duration from 10 minutes to 15 minutes and lasting a total of two hours. Data from each session was collected for aggregation, yielding $n = 10$ values for each metric. The following metrics were computed and aggregated:

- **Incoming Bandwidth Consumption (IBC):** This is computed by dividing the total amount of incoming data by the duration of the measurement session. The domain of this metric is $[0, IBC_M]$ and the metric

is measured in Mbps, where the value IBC_M is the nominal maximum bandwidth declared by the ISP.

- **Incoming Traffic Variation (ITV):** This measures the bandwidth variance for sessions. For a session i, ITV is given by:

$$ITV = \frac{IBC_i - IBC_{i-1}}{length_i}$$

where IBC_i is the incoming bandwidth consumption measured in the i-th session and $length_i$ is the duration of the session. The domain of this metric is $[-ITV_M, ITV_M]$ and the metric is measured in Mbps^2 where

$$ITV_M = \max_i \frac{IBC_M}{length_i}.$$

- **Traffic Tolerance (TT):** This specifies the round trip time (RTT) of an IP packet traveling between the end user's node and the ISP's recursive resolver. The domain of the metric is $[0, +\infty]$ and the metric is measured in seconds.

- **Stub Resolver Cache Poisoning (CP):** This specifies the percentage of poisoned entries in the cache. The domain is $[0, 100]$ with every entry in the cache being verified against a set of trusted recursive resolvers.

- **DNS Requests per Second (DNSR):** This is the total number of DNS queries in a session. The domain is $[0, +\infty]$.

- **Rate of Repeated Queries (RRQ):** This is the number of repeated DNS queries in a session. Under normal conditions, a name is resolved only once due to DNS caching. Many DNS queries for the same name during the same session could be an indicator of malicious activity. The domain is $[0, +\infty]$.

The following set of quality mapping functions for the metrics are employed:

- **Incoming Bandwidth Consumption (IBC):** The quality mapping $q\colon [0, IBC_M] \rightarrow [0, 1]$ for the IBC metric is defined as:

$$q(x) = 1 - \frac{x}{IBC_M}$$

where IBC_M is the maximum bandwidth value provided by the ISP.

- **Incoming Traffic Variation (ITV):** The quality mapping $q\colon [-ITV_M, ITV_M] \rightarrow [0, 1]$ for the ITV metric is defined as:

$$q(x) = 1 - \frac{|x|}{ITV_M}.$$

Table 2. Measurements and quality ratings for Sessions 1, 2 and 3.

	IBC		**ITV**		**TV**		**CP**		**DNSR**		**RRQ**	
	Mbps	**q**	**Mbps²**	**q**	**s**	**q**	**%**	**q**	**#**	**q**	**#**	**q**
S_1	11.8	0.998	0	1	0.80	0.80	9.96	0.90	0.87	1	0.84	0.84
S_2	11.9	0.997	0.0054	0.999	0.74	0.74	6.67	0.93	0.33	0	0.89	0.79
S_3	13.9	0.997	0.0002	0.999	0.78	0.78	10.40	0.89	0.24	0	0.74	0.74

- **Traffic Tolerance (TT):** The quality mapping $q\colon [0, +\infty] \to [0, 1]$ for the TT metric is defined as:

$$q(x) = \begin{cases} 1 & x \leq RTT_{Av} \\ -\frac{x}{RTT_{Av}} + 2 & RTT_{Av} < x \leq 2RTT_{Av} \\ 0 & x > 2RTT_{Av} \end{cases}$$

  where RTT_{Av} is the average RTT value during the session.

- **Cache Poisoning in the Stub Resolver (CP-SR):** The quality mapping $q\colon [0, 100] \to [0, 1]$ for the CP-SR metric is defined as:

$$q(x) = 1 - \frac{x}{100}.$$

- **DNS Requests per Second (DNSR):** The quality mapping compares the current DNS behavior against a previous reference. The quality mapping q for the DNSR metric is defined as:

$$q(x) = \begin{cases} 1 - \frac{x}{2 \cdot DNSR_{Av}} & 0 \leq x \leq 2DNSR_{Av} \\ 0 & x > 2DNSR_{Av} \end{cases}$$

  where $DNSR_{Av}$ is the average number of the DNS requests per second during the session.

- **Rate of Repeated Queries (RRQ):** The quality mapping q for the RRQ metric is defined as:

$$q(x) = 1 - \frac{x}{R_M}$$

  where R_M is the maximum number of DNS requests in the current session. Note that R_M changes for different sessions.

4.2 Aggregation and Experimental Results

Table 2 shows the measurement values and related quality ratings for the experiment. For brevity, data for Sessions 4 through 10 are not presented.

Table 3. Session 1 aggregate results.

	IBC q = 0.998	ITV q = 1	TV q = 0.801	CP q = 0.9	DNSR q = 1	RRQ q = 0.842	Aggregate Result
TE	0.19	0.19	0.19	0.05	0.19	0.19	0.927
PI	0.00	0.00	0.00	1.00	0.00	0.00	0.900
DoS	0.20	0.20	0.20	0.00	0.20	0.20	0.928
Net	0.33	0.33	0.33	0.00	0.00	0.00	0.932
SR	0.00	0.00	0.00	0.12	0.44	0.44	0.918

For every session, the quality ratings of the metrics are aggregated for the end user PoV indices. Table 3 presents the following aggregate results for the first session:

- **Total Evaluation Index (TE):** This provides a global assessment of the PoV aggregated over all considered metrics.
- **Protocol Issues Index (PI):** This estimates possible DNS protocol problems. The index is related to the cache poisoning metric.
- **Denial of Service Index (DoS):** This evaluates how improbable DoS is in a given scenario. The DoS index aggregates all the metrics except for cache poisoning.
- **Net Index (Net):** This estimates the performance of the network component. The Net index aggregates incoming bandwidth consumption, incoming traffic variation and traffic tolerance.
- **Stub Resolver Index (SR):** This evaluates stub resolver performance. The SR index aggregates cache poisoning, DNS requests variation per second and rate of repeated queries.

The final result of each aggregated index for the end user PoV is computed as the average of the results over all ten sessions. The variances are computed to provide estimates of the uncertainty of the results. Figure 3 shows the final values.

4.3 Discussion

The total evaluation index is the primary consideration for the end user PoV – it reflects the overall DNS health using components that can be measured by end users. In the investigated scenario, minor disruptions are deemed to be acceptable (e.g., temporary DNS failures that require the reloading of web pages). For this reason, total evaluation index values less than one are acceptable in a properly functioning system. With the MeNSa framework, it is possible to quantify service levels and to verify if SLOs are violated. As an example, in our experiment, the total evaluation value was computed to be 0.833 with an uncertainty value ± 0.134. Such a value quantifies DNS health as perceived by the

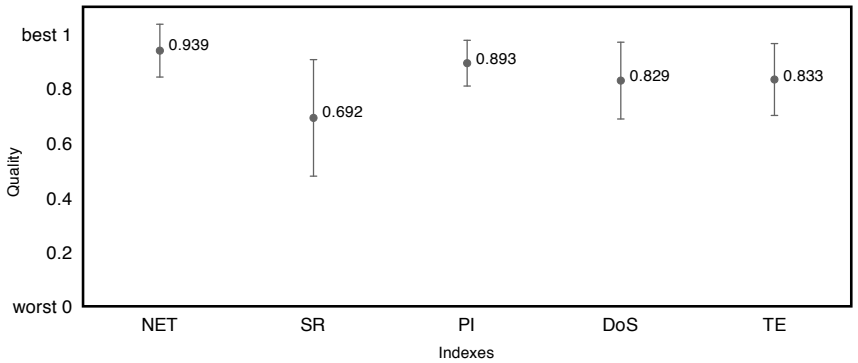


Figure 3. Aggregated results and index uncertainty over all sessions.

end user. The other aggregated results provide insight into the performance of different aspects of the system. This information is valuable because it can help identify components that require further scrutiny in the event of malfunctions.

Our calculations show that the stub resolver is the component that has the highest likelihood to have problems; this is because the stub resolver index value of 0.692 is far from one. In contrast, the Net component evaluation is 0.939 and has a low degree of uncertainty. It is important to note that further analysis is possible if the aggregated results of the recursive resolver PoV are available as an input metric for the end user PoV. In other words, the outputs of a PoV can be used as input metrics to another PoV to provide results with finer granularity. Aggregating PoV values with available local metrics increases the accuracy of an overall assessment and refines the evaluation of single components.

Our investigation also focused on threat scenarios that could affect a targeted infrastructure. Indeed, some of the results provide insights into the likelihood of certain threats or attacks. For example, the high values of the protocol issues and denial of service indices (0.893 and 0.829, respectively) indicate, with a high degree of certainty, that the system was not affected by protocol issues or denial-of-service attacks during the measurement period.

It is important to note that the results presented above cannot be generalized and must be validated using larger sets of experiments. Nevertheless, the study demonstrates the ability to measure and aggregate DNS health metrics.

5. Conclusions

DNS is a critical component of the Internet. Indeed, without DNS services the majority of Internet applications would not function properly. The increasing use of information and communications technologies in critical infrastructure assets makes it vital to protect DNS – targeted attacks that degrade DNS could cause serious consequences to modern society.

The MeNSa framework provides a formal methodology for evaluating DNS health based on requirements identified by the DNS community. The experi-

mental results demonstrate that end user metrics can be aggregated to verify the level of service and identify potential threats. Our future research will continue our efforts at validating the MeNSa framework using larger data sets and also expand the framework to consider other points of view.

References

[1] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, DNS Security Introduction and Requirements, RFC 4033, 2005.

[2] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, Resource Records for the DNS Security Extensions, RFC 4034, 2005.

[3] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, Protocol Modifications for the DNS Security Extensions, RFC 4035, 2005.

[4] S. Bellovin, Using the domain name system for system break-ins, *Proceedings of the Fifth USENIX UNIX Security Symposium*, p. 18, 1995.

[5] E. Casalicchio, M. Caselli, D. Conrad, J. Damas and I. Nai Fovino, Framework Operation, the Web User PoV, Technical Report, Version 1.1, Global Cyber Security Center, Rome, Italy, 2011.

[6] E. Casalicchio, M. Caselli, D. Conrad, J. Damas and I. Nai Fovino, Reference Architecture, Models and Metrics, Technical Report, Version 1.5, Global Cyber Security Center, Rome, Italy, 2011.

[7] E. Casalicchio, D. Conrad, J. Damas, S. Di Blasi and I. Nai Fovino, DNS Metric Use Cases, Technical Report, Version 1.0, Global Cyber Security Center, Rome, Italy, 2011.

[8] S. Castro, D. Wessels, M. Fomenkov and K. Claffy, A day at the root of the Internet, *ACM SIGCOMM Computer Communication Review*, vol. 38(5), pp. 41–46, 2008.

[9] D. Danchev, Hackers hijack DNS records of high profile New Zealand sites, *ZDNet*, San Francisco, California, April 21, 2009.

[10] D. Eastlake, Domain Name System Security Extensions, RFC 2535, 1999.

[11] D. Eastlake and C. Kaufman, Domain Name System Security Extensions, RFC 2065, 1997.

[12] Global Cyber Security Center, The Measuring Naming System Project, Rome, Italy (`www.gcsec.org/activity/research/dns-security-and-stability`), 2012.

[13] D. Goodin, Cache-poisoning attack snares top Brazilian bank, *The Register*, April 22, 2009.

[14] I. Nai Fovino, S. Di Blasi and A. Rigoni, The role of the DNS in the secure and resilient operation of critical infrastructures: The energy system example, presented at the *Sixth International Conference on Critical Information Infrastructure Security*, 2011.

[15] Internet Corporation for Assigned Names and Numbers, Root Server Attack on 6 February 2007, DNS Attack Factsheet 1.1, Los Angeles, California, 2007.

[16] Internet Corporation for Assigned Names and Numbers, Response to Recent Security Threats, ICANN Technical Report, Los Angeles, California, 2008.

[17] Internet Corporation for Assigned Names and Numbers, Security, Stability and Resiliency of the Domain Name System, ICANN Technical Report, Los Angeles, California, 2009.

[18] Internet Corporation for Assigned Names and Numbers, Measuring the Health of the Domain Name System, Report of the Second Annual Symposium on DNS Security, Stability and Resiliency (Kyoto, Japan), Los Angeles, California, 2010.

[19] D. Kaminsky, It's the end of the cache as we know it, presented at *Black Hat USA*, 2008.

[20] B. Krebs, Hackers hijacked large e-bill payment site, *Washington Post*, December 3, 2008.

[21] R. Liston, S. Srinivasan and E. Zegura, Diversity in DNS performance measures, *Proceedings of the Second ACM SIGCOMM Workshop on Internet Measurment*, pp. 19–31, 2002.

[22] E. Mills, Puerto Rico sites redirected in DNS attack, *CNET*, San Francisco, California, April 27, 2009.

[23] M. Santcroos and O. Kolkman, DNS Threat Analysis, Technical Document 2006-SE-01 version 1.0, NLnet Labs, Amsterdam, The Netherlands, 2007.

[24] Y. Sekiya, K. Cho, A. Kato and J. Murai, Research of method for DNS performance measurement and evaluation based on benchmark DNS servers, *Electronics and Communications in Japan (Part I: Communications)*, vol. 89(10), pp. 66–75, 2006.

[25] P. Vixie, DNS and BIND security issues, *Proceedings of the Fifth USENIX UNIX Security Symposium*, p. 19, 1995.

[26] P. Vixie, G. Sneeringer and M. Schleifer, 21 Oct 2002 Root Server Denial of Service Attack – Report, Technical Report, Internet Systems Consortium, Redwood City, California, 2002.

Chapter 13

EMERGENCY MESSAGES IN THE COMMERCIAL MOBILE ALERT SYSTEM

Paul Ngo and Duminda Wijesekera

Abstract The U.S. Department of Homeland Security initiated the Commercial Mobile Alert System (CMAS) to ensure that emergency situations are effectively communicated to the general public. CMAS uses the existing commercial telecommunications infrastructure to broadcast emergency alert text messages to all mobile users in an area affected by an emergency. One of the limitations of CMAS is that the maximum message size is 90 characters of plaintext. This paper proposes an enhancement to CMAS that provides more detailed information within the 90-character text using an encoding technique. The viability of the enhancement is demonstrated using a prototype that generates and broadcasts CMAS emergency alerts to Android phones, on which an emergency response application intercepts, decodes and displays the alerts to users.

Keywords: Commercial Mobile Alert System, emergency response, alert messages

1. Introduction

Protecting assets against man-made and natural emergencies is a priority. However, due to the unexpected nature of emergencies, preparing for, responding to and recovering from an emergency are always challenging. Indeed, the emergency problem space is often overlooked until an emergency arises, often unexpectedly.

In the aftermath of the September 11, 2001 terrorist attacks, communications were identified as a major bottleneck for emergency and rescue operations. Telecommunications service providers experienced an extremely high overload of calls in and out of the stricken areas, which caused congestion at access and core networks, resulting in many calls being blocked or rejected [8]. However, mobile users were still able to send and receive text messages.

In 2006, the U.S. Federal Government established the Worker Adjustment and Retraining Notification (WARN) Act that supported research and develop-

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 171–184, 2012.

ment efforts related to the Common Mobile Alert System (CMAS) [3]. CMAS utilizes the existing commercial telecommunications infrastructure to broadcast emergency alerts and warnings to a specified geographic area. The current messaging protocol standard, however, is limited to 90 plaintext characters, which is not enough to communicate detailed information. This paper describes an enhanced encoding technique that enables the broadcasting of more detailed information while satisfying the 90-character constraint.

2. CMAS Limitations

In 2006, the U.S. Government initiated the Commercial Mobile Alert Service (CMAS) to broadcast emergency alert text messages to the public [5]. Unlike the short message service (SMS) point-to-point communications protocol, CMAS uses a dedicated broadcast control channel to send text message alerts, which can reach millions of wireless subscribers within minutes. Note that CMAS does not require subscriber registration; the service is available as long as a user is within range of a cellular access point. While CMAS is designed to communicate information to the general public during emergency situations, it inherits the following weaknesses from the cellular broadcast service:

- CMAS alert messages cannot broadcast to an area smaller than a cell site, which is defined in the Federal Information Processing Standard (FIPS) code [4]. The area of a cell site varies depending on the population density and can be too large for targeted broadcast alerts in a small-scale emergency (e.g., a burning building or an apartment gas leak).

- CMAS disseminates three types of alerts: (i) Presidential alerts; (ii) imminent threat alerts; and (iii) AMBER alerts [11]. CMAS is not designed to broadcast alerts for local emergencies.

- The CMAS specification [2] states that the Common Alerting Protocol (CAP) version 1.2 is to be used to communicate emergency alerts. However, CAP 1.2 was designed for department and agency communications across different levels of government (e.g., federal, state and local). Also, most of the information in a CAP 1.2 message is not relevant to emergency mobile broadcasting and the message structure does not meet the requirements associated with local emergencies.

- CMAS broadcast messages are limited to 90 characters of plaintext [2]. This size limitation restricts the ability to disseminate detailed and informative emergency messages.

The first three limitations are addressed by our ERApp emergency application for the Android mobile platform [7]. ERApp filters CMAS messages based on the GPS location and displays alerts only if a user is within the affected area. We also introduced the Emergency Alert System (ERAlert) to generate CMAS alerts specifically for local emergencies. Additionally, we suggested enhancements to the CAP 1.2 message structure by adding XML tags to enable

relevant communications. This paper addresses the fourth limitation by employing an encoding scheme that enables detailed information to be sent in a 90-character message.

3. CMAS Enhancement

This section describes the CMAS enhancement for encoding and delivering detailed emergency information messages. The solution affords the flexibility to tailor messages specific to emergency situations.

3.1 ERApp Considerations

In 2003, OASIS sponsored the CAP initiative to provide messaging protocols that facilitate inter-agency emergency communications. CAP, however, is intended to facilitate communications between emergency systems and operators; it was not intended for one-way broadcast alerts to a population. For example, the sender ID field is not relevant to users who receive broadcast alert messages.

The GSM/UMTS cellular broadcast service technical specification does not address broadcast alert messages for an area smaller than a cell site [1, 2]. The ERApp solution, however, overcomes this challenge by enhancing the CAP 1.2 message structure [6] to include three additional tags: (i) affected area; (ii) spreadable; and (iii) location. Broadcast localization is achieved by intercepting and filtering a CMAS cellular broadcast service alert message based on the distance between the location of the emergency and the recipient.

To support the ERApp implementation, the CAP 1.2 message structure must be expanded to accommodate XML tags and values for the additional information [9, 10]. We propose an encoding scheme that enables more emergency data to be placed within the 90-character block. Upon receiving the CMAS broadcast alert message, ERApp decodes the enhanced XML message into its original format and displays the emergency information to the recipient.

3.2 CMAS Architectural Enhancement

ERApp requires a "codepage" to decode an emergency alert message. The codepage contains a list of emergency message formats and region-specific emergency tag repositories, which can be identified by the namespace of the unique XML schema. The unique uniform resource identifier (URI) or uniform resource name (URN) in the namespace is contained in the emergency XML alert message. Note that each emergency tag repository contains a location-specific list of emergency name and value pairs. The codepage can be downloaded automatically to a mobile device during handover (i.e., the process of transferring an ongoing call or data session from one cellular network to another). Enabling codepage download during handover requires a minor enhancement to the current architecture. To simplify our discussion, we consider the Global System for Mobile Communications (GSM). Other networks such as Code Division Mul-

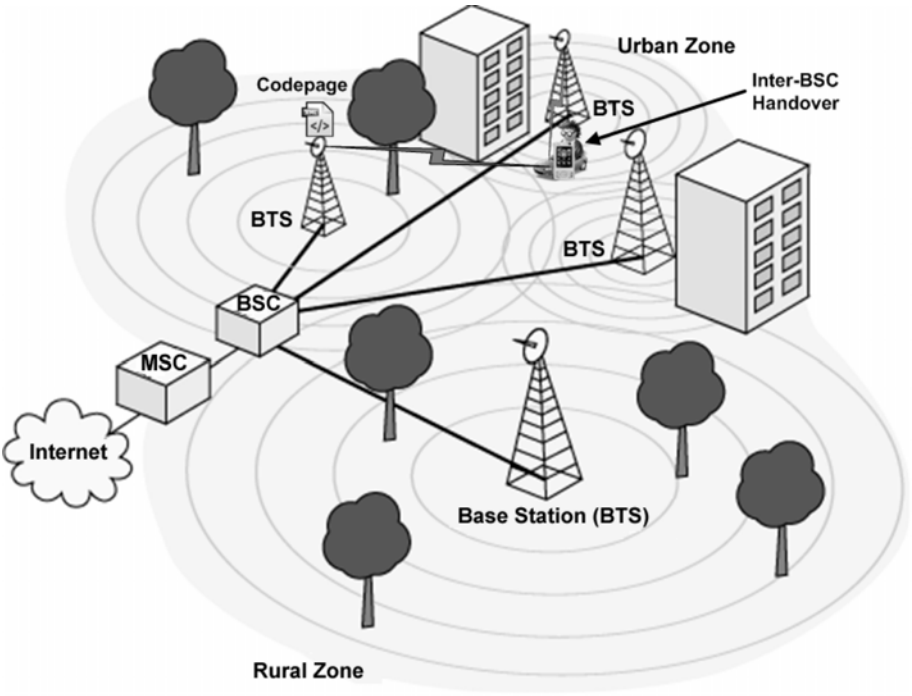


Figure 1. Handover enhancement.

tiple Access (CDMA), Universal Mobile Telecommunications System (UMTS) and Long Term Evolution (LTE) have similar handover procedures.

GSM has four forms of handover: (i) intra-BTS handover; (ii) inter-BTS intra-BSC handover; (iii) inter-BSC handover; and (iv) inter-MSC handover. Although each form has different implementation details, the synchronization procedure between the mobile station (MS) and the base transceiver station (BTS) is common for all four handovers. During synchronization, the codepage is transmitted from the BTS to the MS. Figure 1 shows a user with an Android phone driving from an urban area to a rural zone. When the inter-BSC handover occurs, the new codepage from the rural area is sent to the user's Android phone during the synchronization process.

A second approach is to request the codepage based on the GPS location at the time of ERApp installation. This enables codepage download during a non-emergency when bandwidth may be more readily available. This approach is also better suited for fixed cellular devices that are more likely to remain in one designated cell.

To enhance messaging details in the available 90-character text, we propose two encoding/decoding methods for inclusion with ERApp: (i) predefined method; and (ii) just-in-time method. The predefined encoding method requires ERAlert and ERApp to use the message format specified in the codepage,

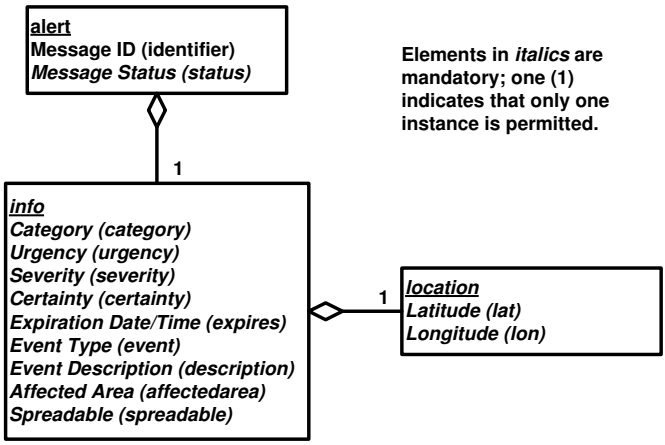


Figure 2. CAP object model for CMAS messages.

which is communicated prior to the broadcast of an alert message. A major advantage of this method is that more bytes are available for emergency data because the tag and attribute names are not encoded. The AMBER alert is a good candidate for the predefined encoding method. However, the predefined method suffers from a loss of flexibility with regard to including or excluding tags or attributes associated with a specific emergency.

The just-in-time encoding method does not require predefined message formats, provided that the alert message complies with the emergency tag repository identified in the codepage. Although the just-in-time method requires more bytes than the predefined method for the tag and attribute names, it offers the flexibility required for specific emergency alerts.

3.3 CMAS Encoding Schemes

Figure 2 presents the CAP object model for CMAS messages. Current encoding schemes limit the utility of the static data fields in CMAS messages. The enhanced structure, however, expands the XML schema to include more expressive information for emergency reporting. To realize the full benefit of the expanded structure, an encoding scheme is required that maximizes the amount of information that can be expressed in each element. Before describing the proposed enhancement, we review the WBXML and prime power encoding schemes currently used by CMAS.

3.3.1 WBXML Encoding. The WBXML encoding converts XML tags and attributes to the associated byte representation. WBXML encodes one byte for a beginning tag, one byte for a value and one byte for an ending tag. A similar method is used to encode an attribute. The details of the WBXML encoding are specified in Algorithm 1.

Algorithm 1 : WBXML Encoding Algorithm (Input: XML Stream)

```
Require: xmlStream ≠ null
 1: tokenSteam← new ByteArrayOutputStream()
 2: handler← new WBXMLContentHandler()
 3: reader← XMLReader.createXMLReader()
 4: reader.setContentHandler(handler)
 5: xmlSource← new InputSource(xmlStream)
 6: reader.parse(xmlSource)
 7: tokens← handler.getTokens()
 8: while tokens.hasNext() do
 9:    aToken← (Token)tokens.next()
10:    tokenStream.write(aToken.getValue())
11: end while
12: return tokenStream
```

The WBXML algorithm requires a valid XML stream as its input. In Line 1, a new token stream is created as a new byte array output stream object to store the encoded values. A WBXML handler is then created that implements callback methods (e.g., beginning tag, closing tag and text value) and loads the codepage based on the namespace specified in an XML alert message. The algorithm creates the XML reader in Line 3 and registers the WBXML handler with the XML reader in Line 4. In Line 5, the algorithm creates a new input source object from the XML stream, which is provided as input for the reader.parse method in Line 6. The reader.parse method evaluates tags and attributes in pre-order. When the parser identifies a beginning or closing tag name, it calls the appropriate callback methods defined in the handler to identify the tag name in the tag repository. The handler.getTokens call in Line 7 returns the list of byte tokens to be written into the token stream. Lines 8 through 11 recursively evaluate the list of tokens and write out the encoded byte values to the output token stream, which is returned in Line 12. This technique is useful for encoding values of known tags and attributes.

3.3.2 Prime Power Encoding.

The prime power encoding method encodes an XML document as a single, albeit large, integer. Note that significant CPU power and computational time may be required to encode and decode a simple XML document. The details of the prime power encoding are specified in Algorithm 2.

Like the WBXML algorithm, the prime power algorithm requires a valid XML stream as input. In Line 1, the algorithm creates the XML prime power content handler. The handler implements the callback functions and loads the codepage based on the namespace specified in an XML alert message. In Line 2, the algorithm creates the reader. The handler is then registered with the reader in Line 3 so callback functions are referenced when XML tags are encountered. In Line 4, xmlSource is created from xmlStream. In Line 5, the reader.parse method performs post-order lookups from the leaf node to the root

Algorithm 2 : Prime Power Encoding Algorithm (Input: XML Stream)

Require: xmlStream $\neq$ null
1: handler← new XMLPPContentHandler()
2: reader← XMLReader.createXMLReader()
3: reader.setContentHandler(handler)
4: xmlSource← new InputSource(xmlStream)
5: reader.parse(xmlSource)
6: **return** handler.getPPValue()

node. The first three prime numbers are reserved for the default node, internal node and leaf node. At every node and node value, the parser performs the prime encoding by taking the smallest available prime and raising it to the power of the integer value represented in the tag repository. Finally, in Line 6 the handler returns the encoded integer value corresponding to the XML document.

Consider the severity tag in the tornado example in Figure 3. The severity tag has an integer value of 2 and a severity value of 2. To encode the severity tag, the parser first encodes the severity tag value. Because the severity value is the leaf node, the parser raises the leaf prime number 3 to the power of 2, yielding a value of 9. The parser continues to encode the severity tag by raising the internal node prime of 2 with the tag integer value of 2. Therefore, the severity tag has the integer value of 4. To complete the encoding, the parser multiplies the tag integer value of 4 with 1953125, which is the result of raising the next available prime of 5 to the tag value integer of 9. The prime power encoding for the severity tag yields the large integer value of 7812500. Note that the integer values can be extremely large and become unmanageable very rapidly.

3.3.3 CMAS Encodings. In order to provide more meaningful CMAS alert messages for XML trees, we introduce the just-in-time and predefined XML encoding schemes. These encoding schemes significantly reduce the number of required encoded bytes.

Just-in-Time Encoding. The just-in-time encoding algorithm has two phases: (i) preprocessing phase; and (ii) encoding phase. The preprocessing phase builds the XML tag repository from the XML schema. For each level of depth in the XML tree, the preprocessing phase examines all possible tag names and associates each unique instance with an integer value starting at 0. The process is repeated for all tag values, attribute names and attribute values.

The preprocessing phase generates a codepage for all region-specific tag repositories. Each tag repository is identified and retrieved according to the unique namespace of the XML schema. The generated codepage is location-specific and pertains to emergencies that occur regularly in the associated region. For example, the codepage for areas in Florida can describe hurricane and

```
<?xml version = "1.0" encoding = "UTF-8"?>
<alert xmlns = "urn:oasis:names:tc:emergency:cap-cmas">
   <identifier>CMAS-01</identifier>
   <status>Actual</status>
   <info>
      <category>Met</category>
      <urgency>Expected</urgency>
      <severity>Severe</severity>
      <certainty>Observed</certainty>
      <expires>2010-10-02T17:00:00-0500</expires>
      <description>Multiple tornados are expected
          around 2PM in the Washington, DC area.
      </description>
      <affectedarea>1000</affectedarea>
      <spreadable>No</spreadable>
      <event>Tornado</event>
      <location>
         <lat>38.882334</lat>
         <lon>-77.171091</lon>
      </location>
   </info>
</alert>
```

Figure 3. CMAS mobile alert message for a tornado warning.

tornado related tags, while the codepage for areas in California can describe earthquake and wildfire tags.

Algorithm 3 : Preprocessing Algorithm (Input: XML Emergency Schema)

Require: xmlSchemaStream $\neq$ null
1: depth $\leftarrow$ getDepth(xmlSchemaStream)
2: maxTags $\leftarrow$ getMaxTags(xmlSchemaStream)
3: numTagBits $\leftarrow$ log_2(maxDepth) + log_2(maxTags) + 1
4: maxTagValues $\leftarrow$ getMaxTagValues(xmlSchemaStream)
5: maxAttrValues $\leftarrow$ getMaxAttrs(xmlSchemaStream)
6: maxValues $\leftarrow$ max(maxTagValues, maxAttrValues)
7: numValueBits $\leftarrow$ log_2(maxValues) + 1
8: numEncodingBits $\leftarrow$ max(numTagBits, numValueBits)
9: encodingScheme $\leftarrow$ getEncodingScheme(numEncodingBit)
10: **return** encodingScheme

As shown in Algorithm 3, the preprocessing phase uses the depth and the maximum number of tags, tag values, attribute names and attribute values. Furthermore, the preprocessing phase examines all static value tags and consolidates them to fit the selected encoding schema. Note that preprocessing generates a combined tag in the tag repository, which significantly reduces the number of encoded bytes.

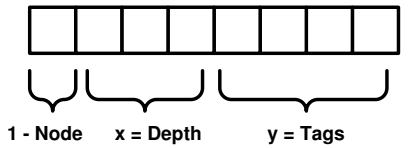


Figure 4. Node depth and tags.

The just-in-time encoding algorithm (Algorithm 3) requires a valid XML emergency schema as input. All the nodes in the schema are examined in Line 1 to compute the depth and in Line 2 to determine the maximum number of tags (maxTags). Line 3 computes the number of bits required to encode the tags. The algorithm then computes the maximum number of tag values (maxTagValues) in Line 4 and the maximum number of attribute names and attribute values (maxAttrValues) in Line 5. The maximum value (maxValues) between the two is computed in Line 6. The algorithm then computes the number of bits to encode the values. The number of bits required for encoding the XML schema is computed in Line 8. In Line 9, the encoding scheme (encodingScheme) is determined from one of the four values (e.g., 8 bits, 16 bits, 32 bits or 64 bits). The encoding is returned in Line 10.

An application of the just-in-time encoding scheme is illustrated in Figure 4. As an example, a one-byte encoding scheme is used to encode an XML document with depth $x = 8$ and maxTags $y = 16$. The number of encoding bits (numTagBits) z_1 is computed as:

$$z_1 \geq log_2(x) + log_2(y) + 1$$

where x is the depth and y is the maximum number of tags at each depth.

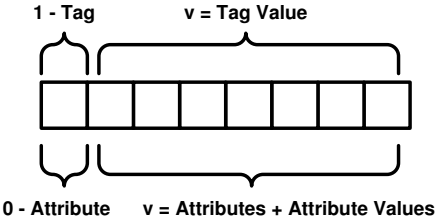


Figure 5. Attribute depth and tags.

Figure 5 illustrates the encoding of a tag value, attribute name and attribute value. Note that the depth does not have to be encoded because it is incorporated into the node. To compute the number of encoding bits for the values (numValueBits) z_2, the maximum number of tag values (maxTagValues) and the maximum number of attributes and attribute values (maxAttrValues) must be determined first. The number of encoding bits for the values is computed

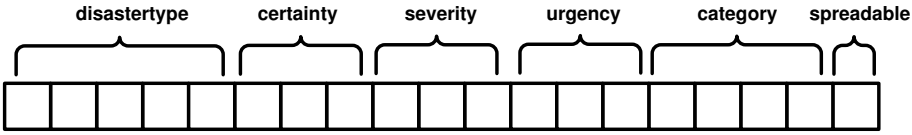


Figure 6. Combined tag.

as:

$$z_2 \geq log_2(v) + 1$$

where v is the maximum number between tag values, attribute names and attribute values for each tag (maxValues). Finally, the number of encoding bits (numEncodingBits) z required is given by:

$$z = max(z_1, z_2).$$

In Figure 5, we use an 8-bit encoding scheme to encode a known tag or a known attribute. With this encoding scheme, the maximum number for tag values, attribute names and attribute values is 128 for each tag.

Consider, for example, the tornado alert message discussed previously (Figure 3). The category, urgency, severity, certainty and spreadable tags have static values. If each tag is encoded separately, ten bytes are required to encode the five tags and five tag values. As shown in Figure 6, combining the tags realizes an encoding that uses only three bytes. Note that the category tag has twelve possible values and requires four bits, whereas the spreadable tag is a Boolean value and only requires one bit.

Algorithm 4 : MXML Encoding Algorithm (Input: XML Stream)

Require: xmlStream $\neq$ null
1: xmlStream← combinedTags(xmlStream)
2: handler← new MXMLContentHandler()
3: reader← XMLReaderFactory.createXMLReader()
4: reader.setContentHandler(handler)
5: xmlSource← new InputSource(xmlStream)
6: reader.parse(xmlSource)
7: masterNode← handler.getMasterNode()
8: outputStream← new ByteArrayOutputStream()
9: masterNode.writeStream(outputStream)
10: arrayBytes← outputStream.toByteArray()
11: outputStream.close()
12: **return** arrayBytes

After the preprocessing phase, the XML alert message document is ready for the encoding phase. Algorithm 4 specifies the CMAS encoding scheme, which we refer to as the mobile XML (MXML) encoding.

The MXML algorithm requires the codepage and a valid XML stream as input. In Line 1, combined tags in the tag repository generate a new XML

stream. The MXMLContentHandler function in Line 2 creates the handler, which implements the callback functions and loads the codepage based on the XML namespace specified in the XML alert message. In Line 3, the XMLReader function creates the reader and, in Line 4, the handler is registered with the reader. In Line 5, xmlSource is created from xmlStream. In the encoding phase, the MXMLContentHandler creates the internal tag repository. The reader.parse method is called in Line 6 with xmlSource as the parameter; the reader examines each XML tag value, attribute name and attribute value in pre-order. The parser then encodes the XML document, starting from the root element at depth zero and continues recursively to the other elements.

The encoding method uses a pre-order traversal to encode every tag name, tag value, attribute name and attribute value. For every value encountered by the parser, the value in the tag repository is correlated based on the depth level and its parent node. The value is encoded by inserting the byte representation into a byte array. The encoding bytes in the array from the child elements are appended to the parent node. If the value is not statically known (i.e., the value is not registered in the tag repository), then it is encoded as a text value with null bytes representing the beginning and end of the text. After the encoding phase is complete, a byte array stream containing the encoding bits is returned. The encoding byte array stream must be converted into an array of characters using Base64 encoding to provide human-readable text. The additional message length provided by the encoding is given by:

$$Base64\ Length = (Bytes + 2 - ((Bytes + 2)\ mod\ 3))/3 * 4.$$

An additional consideration is that the cellular broadcast service uses an independent broadcast control channel with dedicated bandwidth to send emergency alerts. Because there is no competition with other channels for bandwidth, emergency information can be segmented into multiple messages if the alert exceeds the standard 90 characters. ERApp uses the message header fields to reconstruct the original emergency information in its entirety.

Predefined Encoding. Similar to the just-in-time encoding algorithm, the predefined encoding algorithm has two phases: (i) preprocessing phase; and (ii) encoding phase. The implementation mirrors the just-in-time algorithm, but with a slight modification in the preprocessing phase. Specifically, in the case of the predefined encoding method, the alert message format is defined and stored in the codepage during the preprocessing phase. ERAlert and ERApp use the prescribed message format to encode and decode the alert message, respectively. The encoding phase proceeds as described for the just-in-time encoding.

4. Experimental Evaluation

In order to evaluate performance, alert information is classified as either static or dynamic. Static data fields contain predefined values that emergency

Table 1. Performance evaluation summary.

Encoding Algorithm	Alert Time	Encoding Length	Base64 Length
WBXML	179 ms	267 bytes	356 bytes
Prime Power	Infeasible	N/A	N/A
MXML (Predefined)	160 ms	98 bytes	132 bytes
MXML (Just-in-Time)	158 ms	118 bytes	160 bytes

operators can select based on the situation; examples include category, urgency, severity, certainty and event. Dynamic data fields contain values that cannot be predefined due to intractable uncertainty; examples include event location, expiration time and description.

We revisit the tornado example to illustrate the encoding of static and dynamic data. The data element, category, represents the static environment and is specified as <category>Met</category>. Let s be the description of the category data field and let $D(s)$ be its length such that the normal value of $D(s)$ is 24 characters.

All the tokens and their values are defined in the tag repository for CMAS alerts. Each token is represented by an integer value starting at zero. The category field and MET values are encoded as 0 and 2 (bytes, not integer values), respectively. Therefore, the CMAS encoding requires only two bytes instead of the eight bytes needed to represent two integers; thus, the corresponding $D(s)$ value is two bytes.

For the dynamic data category, data fields and values are encoded according to their data types. For example, the expiration time may be encoded as a string representing the timestamp, which requires up to 20 characters. However, the data field also can be encoded as a long value for milliseconds or an integer value for seconds. Because the number of milliseconds provides more detail than is necessary, the expiration time is encoded as an integer value that requires only four bytes.

The WBXML, prime power, MXML just-in-time and MXML predefined algorithms were executed for CMAS mobile alert messages corresponding to the tornado example. A Dell Latitude E6400 with dual core processors was used as the computing platform. Because CMAS only allows 90 characters per broadcast, the tornado alert was segmented into two broadcast messages and reconstructed by the receiving ERApp. Table 1 shows the results for the average alert time, encoding length and the Base64 encoding length. The MXML predefined encoding provides the shortest length for readable text (i.e., Base64 encoding length). The MXML just-in-time and MXML predefined encodings used less bytes to encode the message than the original CMAS encoding scheme while also minimizing the alert time. Note that the prime power encoding was declared to be infeasible because it exhausted the CPU utilization rate and was unable to encode the alert message even after several hours of processing.

5. Conclusions

The CMAS extension described in this paper enables detailed emergency information to be incorporated in alert messages while complying with the 90-character message specification. The utility of the approach is demonstrated by the ability to install the ERApp emergency response application on an Android platform and receive detailed emergency alert messages.

References

[1] Alliance for Telecommunications Industry Solutions, Implementation Guidelines and Best Practices for GSM/UMTS Cell Broadcast Service, ATIS-0700007, Washington, DC, 2009.

[2] Alliance for Telecommunications Industry Solutions, Commercial Mobile Alert Service (CMAS) via GSM/UMTS Cell Broadcast Service Specification, ATIS-0700006, Washington, DC, 2010.

[3] Federal Communications Commission, Common Mobile Alert System (CMAS), Washington, DC (`www.fcc.gov/cgb/consumerfacts/cmas.html`), 2011.

[4] National Institute of Standards and Technology, Federal Information Processing Standards Publications, Gaithersburg, Maryland (`www.itl.nist.gov/fipspubs/index.htm`).

[5] National Public Safety Telecommunications Council, Commercial Mobile Alert Service Architecture and Requirements, Version 0.6, Littleton, Colorado (`www.npstc.org/download.jsp?tableId=37&column=217&id=703&file=PMG-0035_Final_Recommendations_v0_6.pdf`), 2007.

[6] P. Ngo and D. Wijesekera, Using ontological information to enhance responder availability in emergency response, *Proceedings of the Semantic Technology for Intelligence, Defense and Security Conference*, 2010.

[7] P. Ngo and D. Wijesekera, Enhancing the usability of the Commercial Mobile Alert System, in *Critical Infrastructure Protection V*, J. Butts and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 137–149, 2011.

[8] Northern Virginia Resource Center for Deaf and Hard of Hearing Persons, Emergency Preparedness and Emergency Communication Access Lessons Learned Since 9/11 and Recommendations, Fairfax, Virginia (`tap.gallaudet.edu/emergency/nov05conference/EmergencyReports/DHHCANEmergencyReport.pdf`), 2004.

[9] Organization for the Advancement of Structured Information Standards, Emergency Data Exchange Language (EDXL) Distribution Element, v1.0, OASIS Standard EDXL-DE v1.0, Burlington, Massachusetts (docs.oasis-open.org/emergency/edxl-de/v1.0/EDXL-DE_Spec _v1.0.pdf), 2006.

[10] Organization for the Advancement of Structured Information Standards, Common Alerting Protocol Version 1.1 (Approved Errata), Burlington, Massachusetts (docs.oasis-open.org/emergency/cap/v1.1/errata/CAP-v1.1-errata.html), 2007.

[11] U.S. Department of Justice, AMBER Alert, Washington, DC (www.amberalert.gov).

IV

INFRASTRUCTURE MODELING AND SIMULATION

Chapter 14

A ONE-DIMENSIONAL SPARSE SPACE-TIME SPECIFICATION OF THE GENERALIZED RAILROAD CROSSING

Michael Gosnell and Bruce McMillin

Abstract Modeling and reasoning about critical infrastructure systems is a complex endeavor. Various calculi and algebras have been crafted to help specify physical properties such as time and space, but these do not always translate well between physical entities and their conceptual specifications. Although real-world critical infrastructure systems involve components of both time and space, many existing specification methods focus most strongly on the temporal components, leaving spatial details largely ignored or forcing then to fit within the confines of the temporal specification. This paper presents a one-dimensional sparse space-time specification created using a spatial-temporal logic in which real-world constraints are incorporated in the logic using the *next* operator. The simplicity and utility of the spatial-temporal formalism is demonstrated by applying it to the generalized railroad crossing problem.

Keywords: Generalized railroad crossing, sparse space-time, assertion checking

1. Introduction

Real-world critical infrastructure systems are susceptible to errors, including hardware malfunctions and failures, software malfunctions and corruption, malicious attacks, and unknown and unseen failures. While many techniques exist for helping mitigate errors, critical infrastructure protection is based on the assumption that the correct operation of the systems of interest is known. Expressing the correct operating behavior of a system can take on many forms, depending on the types of error mitigating techniques and personal preferences.

System specifications can be formulated in a variety of ways, such as using calculi [10], temporal logic [9, 10], or automata or state transition systems [5, 6] that can be automatically verified with model checking. Expressing the

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 187–204, 2012.

Table 1. RCC interval relationships.

Interval	Definition
$C(x, y)$	x connects with y
$DC(x, y)$	x is disconnected from y
$P(x, y)$	x is a part of y
$PP(x, y)$	x is a proper part of y
$EQ(x, y)$	x is equivalent to y
$O(x, y)$	x overlaps y
$DR(x, y)$	x is discrete from y
$PO(x, y)$	x partially overlaps y
$EC(x, y)$	x is externally connected with y
$TPP(x, y)$	x is a tangential proper part of y
$NTPP(x, y)$	x is a nontangential proper part of y

correctness of a critical infrastructure system is often challenging due to the specification requirements. This paper presents a new "sparse space-time" approach, which is designed to better encapsulate physical characteristics within the specification, allowing for more natural specifications of components in the critical infrastructure protection domain.

2. Background

This section discusses the region connection calculus (RCC), which provides qualitative spatial relationships that are used within the specification language. Also, it discusses spatial-temporal logics that help capture temporal aspects. Finally, the generalized railroad crossing (GRC) problem is presented along with high-level definitions of safety and liveness.

2.1 Region Connection Calculus

The region connection calculus (RCC) [11] is an extension of the mereological- and topological-based work of Clarke [2] to form an interval logic for dealing with space. This interval spatial work incorporates qualitative relationships similar to Allen's interval temporal logic [1]. The RCC spatial interval relationships are summarized in Table 1, where $EQ(x, y)$ replaces the original $x = y$ notation. Note that P, PP, TPP and $NTPP$ are not symmetric and, therefore, support inverses, which are denoted by appending $^{-1}$ as in $NTPP^{-1}$.

In addition to the general RCC, a smaller set of jointly exhaustive pairwise disjoint relations are provided. Figure 1 shows these base relations along with the potential transitions between relations. These eight relationships form the basis of the region connection calculus known as RCC-8.

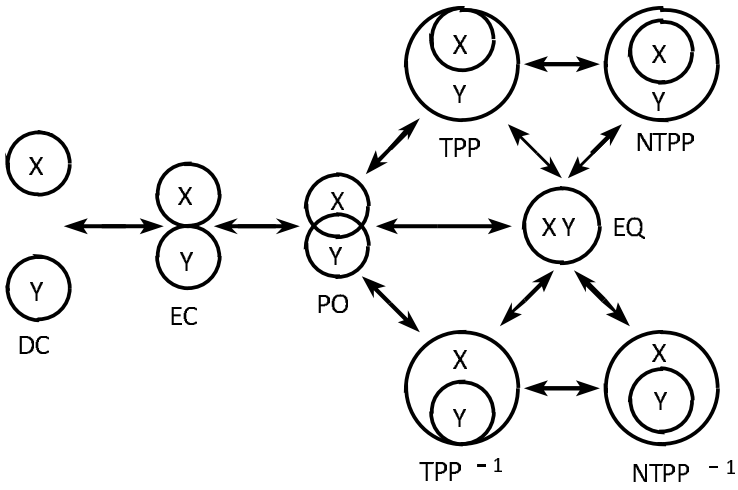


Figure 1. RCC-8 base relations and potential transitions.

2.2 Spatial-Temporal Logics

A spatial-temporal logic (STL) is the combination of a spatial logic with a temporal logic [7]. More specifically, an STL incorporates the expressiveness of the spatial and temporal logics, as well as the interactions between the spatial and temporal components as allowed by the STL. There are some standard characteristics of STLs, e.g., an STL should be able to specify spatial propositions in relation to time. However, the full specification of principles, which is unique to each particular STL, dictates how the spatial-temporal predicates extend individual spatial and temporal propositions and how truth values change over time. For example, the assertion:

$$NTPP(Computer\ Science, Campus) \Rightarrow \bigcirc\, NTPP(Computer\ Science, Campus)$$

states that if the Computer Science building is a (nontangential proper) part of the campus, it will remain so "at the next state" (where "state" might be a time, system state, world, etc.).

Multiple combinations of spatial and temporal logics are presented in [12], where RCC-8 is used as the basis for spatial reasoning. In this paper, STL specifications take the RCC-8 form with branching temporal logic as fits with STL logic $\mathcal{ST}_2$ in [12]. Because the focus is on aspects of specification languages as they relate to runtime assertion checking (and not model checking where issues such as decidability are of importance), additional technical aspects of STL are not included.

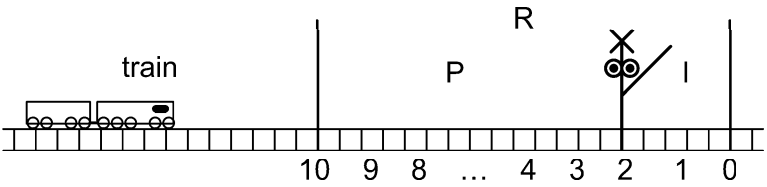


Figure 2. Generalized railroad crossing example.

2.3 Generalized Railroad Crossing

A generalized railroad crossing (GRC) problem is attractive as a basic scenario in a multitude of application domains. A railway can be interpreted (with some basic constraints) primarily as a one-dimensional domain, which reduces the complexity of having to conduct a multidimensional analysis. The railway is a component of the larger transportation infrastructure, and the model can be extended to vehicular, airspace and littoral domains. The GRC problem introduces various spatial and temporal facets dealing with crossing gate behavior and motor vehicle traffic through the crossing. Although many rail crossing problems contain similar components [5, 6, 10], this paper uses the basic syntax described in [6], which is summarized below.

Figure 2 shows a GRC example, where I is the crossing, P is a portion of track before I and R is the region of interest for train r. Trains are denoted r, r', r'', ... as needed. To aid in the discussion of the shape calculus implementation, a discrete numerical representation is added, where the region I corresponds to the interval [0, 2) and P corresponds to the interval [2, 10). The GRC problem provides the opportunity to investigate simple safety and liveness/utility functions that ensure that the gate is down when a train is in the crossing and that the gate returns to the up position when no train is present. These properties are expressed through the specification languages used for model checking or runtime assertions as described in the following sections.

3. Sparse Space-Time

Specification languages come in many varieties (e.g., automata, state transition diagrams and logics) and can be used to model aspects of physical implementations. Often, the physical characteristics of an implementation can introduce additional system constraints that are not traditionally present within current reasoning. For example, vehicles are constrained to specific rates of change (e.g., acceleration/deceleration) and system state sampling dictates what can be observed and what might be missed. In order to incorporate physical property constraints within the specification language, we couple these aspects within the spatial-temporal *next* operator. This approach is inspired by Kopetz's work with dense and sparse time [8] and naturally fits with the system sampling frequency in that each *next* state corresponds to a state capture of the physical system. In Kopetz's original work, real time (dense time) is constrained to

sparse time intervals such that ordering properties can be maintained in a distributed system with respect to allowable global clock drift. In this approach, events occurring in real time (but after a sparse time interval) are recorded in the next sparse time interval. In our work, the true occurrence of events (in real time) can only be observed at discrete system state capture intervals (the equivalent of sparse time).

The goal of sparse space-time is twofold: (i) fit naturally with the sparse temporal frequency of system state collection; and (ii) provide the capability to reason in the sparse spatial domain (RCC) without risking the loss of dense state transitions. For example, when disconnect holds at one system state capture and the spatial relationship transitions to edge connect and then to partially overlapping before the next system state capture, it is necessary to account for the occurrence of both edge connect and partially overlapping relationships. Without a proper mechanism, sparse space-time in our approach, the system may not be able to recognize when a spatial relationship is satisfied.

This concept is not necessarily new. Gerevini and Nebel [3] refer to the general notion as the "continuity constraint." However, extending this baseline concept with the wider breadth of available actions that occur between state captures introduces an opportunity to include implementation constraints in the logic that can abstract away some of the complexity and allow for the more natural expression of assertions. In this way, physical properties can be included within baseline specification pieces, and then separated from the other dynamics of desired system operation. This intuitive mechanism abstracts the physical system constraints in the specification language while maintaining the specific restrictive details.

Definition. The *next* operator in sparse space-time is defined as:

$$\sigma_i \models \mathcal{P}, \;\; \sigma_{i+1} \models \bigcirc \mathcal{P} \tag{1}$$

where σ is the spatial-temporal domain $\mathcal{T} \times \mathcal{S}$ and i denotes the temporal discretization.

In the sparse space-time approach, space is discretized among the included dimensions and the resulting spatial regions can be reasoned about using the discretization. The spatial discretization aspect becomes important with respect to the relationship between the spatial regions of interest, but mostly in the operational semantics regarding spatial-temporal transitions. The complementary aspect of our sparse space-time approach is the temporal discretization. The temporal discretization itself is fairly straightforward and can be coupled to the sparse space concept of sparse time intervals corresponding to system state collections with a sampling speed ($SamplingRate$). The innovative aspect incorporates the physical properties (e.g., sampling rate) with the discretization knowledge through Equation (1) to provide additional system constraints that include limiting the available spatial-temporal transitions that occur during system state collection points.

Wolter and Zakharyaschev [12] note that the "next time" operator makes no sense in dense time flows such as $\{\mathbb{Q}, <\}$ or $\{\mathbb{R}, <\}$. This follows because any two real numbers will have some identifiable real number between them and, therefore, no quantifiable "next" state. The sparse space-time approach follows naturally and extends the notion of density and sparsity into the spatial realm. Shifting from a dense space to a sparse space equivalently shifts the spatial referencing into a uniform metric space, which remains a topological space. Thus, within sparse space-time, sparse space is handled as a metric space on some flow such as $\{\mathbb{N}, <\}$ or $\{\mathbb{Z}, <\}$. Remaining within a topological space, the fundamental spatial-temporal logic work of Wolter and Zakharyaschev [12] continues to hold, and each of the fundamental RCC-8 relationships can be represented in sparse space-time. However, it still remains to be shown whether or not a sparse space-time representation can represent the physical characteristics of a modeled system. In other words, is sparse space-time sound and complete with respect to expressing sparse truths of a dense physical system?

3.1 Sparse Space-Time Soundness

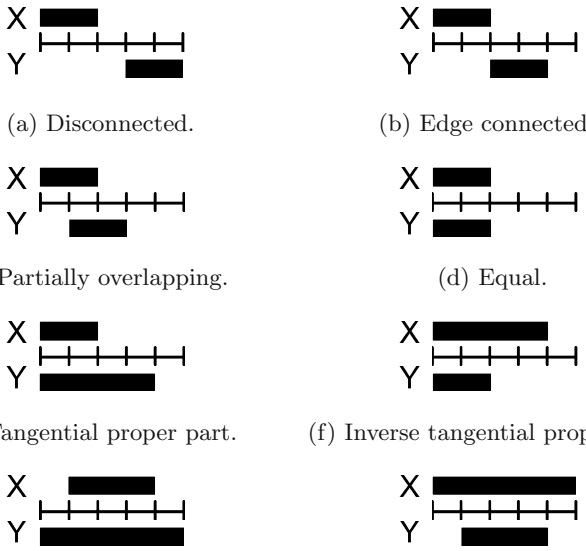
As with any new approach, it is important to understand the benefits and limitations. Logic systems often provide measures of the soundness and completeness as a fundamental baseline. Logical soundness expresses the property that the logic only proves formulas that are valid. The logical soundness of the spatial-temporal logic was addressed by Wolter and Zakharyaschev [12]. Since sparse space-time retains a spatial-temporal logic base and changes only the semantics of the *next* operator, the notion of sparse space-time soundness is taken to mean the ability to express every RCC-8 relationship as specified in Theorem 1 below.

Lemma 1 (Metric Space Uniqueness). *When implemented as a metric space with a simple temporal distance metric, each sparse space-time capture is unique.*
Proof: A simple metric space distance can be defined as:

$$d(p_1, p_2) = \sqrt{(t_2 - t_1)^2}$$

where distance d is calculated as the temporal difference between two points $p_1 = (x_1, y_1, z_1, t_1)$ and $p_2 = (x_2, y_2, z_2, t_2)$. Since the sparse space-time *next* operator is defined in Equation (1) with respect to the temporal discretization i, if $p_1 \neq p_2$, then $t_1 \neq t_2$. With all points unique, $t_1 \neq t_2$ holds for all pairwise comparisons, leading to a strict total ordering under $<$ of all sparse space-time events. □

Theorem 1 (Sparse Space-Time Soundness). *Every one-dimensional spatial-temporal relationship is expressible in one-dimensional sparse space-time.*
Proof: By definition, RCC-8 is jointly exhaustive and pairwise disjoint; this



(a) Disconnected. (b) Edge connected.

(c) Partially overlapping. (d) Equal.

(e) Tangential proper part. (f) Inverse tangential proper part.

(g) Nontangential proper part. (h) Inverse nontangential proper part.

Figure 3. Expressing spatial RCC-8 representations in sparse space-time.

means that all spatial situations can be expressed as one of these relationships. Lemma 1 provides the complementing temporal expression through a total temporal ordering of sparse space-time. This result, coupled with showing that sparse space-time can express all the RCC-8 relationships, demonstrate the soundness of the expression in sparse space-time. Figure 3 expresses all the RCC-8 relationships in sparse space-time; this demonstrates soundness. □

Interval temporal logic (ITL) is a temporal logic designed around intervals of time and relationships between the intervals [1]. ITL expresses qualitative relationships between two intervals of time through seven relations and thirteen interval relationships. The equality (*equal*) relationship is reflexive, which eliminates its dual, yielding thirteen relationships from seven temporal interval relations. Figure 3 shows how one-dimensional RCC-8 representations naturally map to ITL relations: X *before* Y (Figure 3(a)), X *meets* Y (Figure 3(b)), X *overlaps* Y (Figure 3(c)), X *equals* Y (Figure 3(d)), X *starts* Y (Figure 3(e)), Y *starts* X (Figure 3(f)), X *during* Y (Figure 3(g)), and Y *during* X (Figure 3(h)).

In general, multiple ITL relations can map to a single RCC-8 relationship, e.g., X *before* Y and Y *before* X both represent DC(X,Y). Reducing the set of ITL expressions to a subset that represents RCC relationships yields: BEFORE, MEETS, OVERLAPS, EQUAL, STARTS/FINISHES, STARTS^{-1}/ FINISHES^{-1}, DURING and DURING^{-1}. Syntactically, DURING indicates X

(a) Equal objects. (b) Nontangential proper part.

Figure 4. Dense EQ and NTPP mappings to sparse EQUALS.

during Y and DURING^{-1} indicates Y *during* X for DURING, STARTS and FINISHES in the discussion of dense and sparse mappings.

3.2 Sparse Space-Time Completeness

Logical completeness expresses the property that any stated proposition can be proven true or false. For sparse space-time to be complete, any statement expressible in sparse space-time would have to be provable in dense space. Stated another way, the completeness of sparse space-time refers to the ability to map between dense and sparse space. A simple example shows that a $1:1$ mapping is not possible.

Consider a one-dimensional sparse space representation where a sparse spatial discretization is flagged if any part of the dense spatial object occupies a portion of the sparse space. Two dense spatial conditions can result in the same sparse spatial representation as shown in Figure 4. Thus, the sparse spatial representation is insufficient to show a unique dense spatial configuration.

Table 2. Mappings from dense RCC relationships to intervals.

RCC Relationship	**Possible Interval Relationships**
DC	BEFORE, MEETS
EC	MEETS
PO	STARTS^{-1}/FINISHES^{-1}, OVERLAPS
EQ	EQUAL
TPP	STARTS/FINISHES, EQUAL
NTPP	STARTS/FINISHES, EQUAL, DURING
TPP^{-1}	STARTS^{-1}/FINISHES^{-1}
NTPP^{-1}	DURING^{-1}

It is assumed that a region of interest must be at least as large as a single spatial discretization. If this were not the case, any RCC-8 relationship could occur within a spatial discretization, allowing no knowledge of the dense spatial relationship. Working in the single spatial dimension, the sparse space representations map naturally to Allen's temporal interval relationships [1]. Interested readers are referred to [4] for a complete analysis of the mapping along with all the full proofs. Table 2 summarizes all the possible mappings between dense RCC-8 relationships and one-dimensional sparse space-time representations.

Theorem 2 (Sparse Space-Time Completeness). *All dense one-dimensional space-time can be represented in one-dimensional sparse space-time.*
Proof: The proof follows directly from Table 2. □

Lemma 2. *The one-dimensional sparse space-time relationships DURING, DURING^{-1}, BEFORE and OVERLAPS uniquely map back to dense space.*
Proof: The proof follows directly from Table 2. □

Theorem 3 (Transitions). *If it can be shown that the EQ RCC relationship will never hold (e.g., by assuring that spatial objects will never be the exact same size), then the remaining one-dimensional dense space transitions can be captured in sparse space-time.*

Proof: Eliminating the RCC-8 transitions through EQ yields three possible transition paths stemming from PO: (i) between PO and DC; (ii) between PO and NTPP; and (iii) between PO and NTPP^{-1}. From Lemma 2 and Table 2, OVERLAPS uniquely maps back to PO, meaning that OVERLAPS can uniquely identify a one-dimensional dense space PO relationship. Likewise, BEFORE maps to DC, DURING maps to NTPP and DURING^{-1} maps to NTPP^{-1}. Therefore, due to Theorem 2, transitions between any two sparse space-time observations of OVERLAPS, BEFORE, DURING and DURING^{-1} lead to a path that guarantees a dense RCC-8 relationship held between observations. □

From Theorem 3, four sparse space-time states can account for dense states and transitions of seven of the eight RCC-8 relations. While it is not possible to completely map from the sparse space-time domain into dense space, these properties allow for RCC reasoning within one-dimensional sparse space-time.

4. Sparse Space-Time GRC Specification

Having defined sparse space-time, we can now attempt to specify the generalized railroad crossing within the framework. The specification begins by capturing the dynamics of gate operation, because the position of the gate determines if the overall system is safe.

Gate operation can be specified in a number of ways. We treat the gate almost as a separate entity because it has a spatial relationship between its opening and closed positions, which could be thought of as separate from the one-dimension of the rail crossing. Using the notation that *gate* indicates the gate position in radians, where $gate = 0$ indicates that the gate is down and $gate = \frac{\pi}{2}$ indicates the gate is up, a minimum gate speed GS_{min} and maximum gate speed GS_{max} in rev/s can be incorporated to express gate opening and closing. These measurements could be transformed back into the straight line one-dimensional characteristics. However, the circular nature of gate operation

fits more naturally with circular parameters that are perfectly acceptable in the sparse space-time approach.

Given the proper part (PP) specified as:

$$PP(\alpha, \beta) = TPP(\alpha, \beta) \vee NTPP(\alpha, \beta).$$

and constraining the gate to only operate between $gate = 0$ (satisfying the Boolean DOWN) and $gate = \frac{\pi}{2}$ (satisfying UP), the minimum and maximum distances of gate travel are defined using the spatial region extending from:

$$\gamma = gate + \frac{GS_{min} \times 2\pi}{SamplingRate} \quad \text{to} \quad gate + \frac{GS_{max} \times 2\pi}{SamplingRate}.$$

The following specification captures the opening gate operations, beginning with the initial transition:

$$DOWN \wedge \bigcirc \neg DOWN \Rightarrow PP(\bigcirc gate, \gamma) \wedge GoingUp.$$

This equation states that if the gate is changing from the down position, then it must travel at least the distance traveled at the minimum gate speed during the system state capture and no more than the maximum distance.

Additional variables are introduced to ensure that the gate condition is "going up" or "going down" to ensure continuation of motion. The specification to capture the gate movement from DOWN to UP is quite similar:

$$GoingUp \wedge \neg UP \Rightarrow \bigcirc(UP \wedge \neg GoingUp) \vee PP(\bigcirc gate, \gamma).$$

Expressing the gate closing operation follows similarly using:

$$\delta = gate - \frac{GS_{min} \times 2\pi}{SamplingRate} \quad \text{to} \quad gate - \frac{GS_{max} \times 2\pi}{SamplingRate}$$

as the spatial region that the gate will maintain during the following state capture given the speed bounds.

The initial transition is captured as:

$$UP \wedge \bigcirc \neg UP \Rightarrow PP(\bigcirc gate, \delta) \wedge GoingDown$$

with the subsequent motion confined by:

$$GoingDown \wedge \neg DOWN \Rightarrow \bigcirc(DOWN \wedge \neg GoingDown) \vee PP(\bigcirc gate, \delta).$$

A final safety bound constrains the gate to be between the UP and DOWN positions as constant $PP(gate, [DOWN, UP])$, meaning that the gate must always remain somewhere between (but inclusive of) the range $[0, \frac{\pi}{2}]$. This somewhat tedious specification of gate operation now makes it possible to capture much more powerful aspects of the GRC safety and liveness than with more concise equations.

4.1 GRC Safety

As shown in Figure 2, r, P, I and R represent the spatial entities train, pre-crossing, in-crossing and region of interest (where $R = P + I$). The initial safety specification is expressed in spatial-temporal logic as:

$$\neg DC(r, I) \Rightarrow DOWN$$

which requires that the gate be down whenever a train is present in the crossing area I. To obtain this safety property, the entry of the train into region P can trigger the lowering of the gate as in:

$$DC(r, P) \wedge \bigcirc \neg DC(r, P) \Rightarrow GoingDown.$$

This assertion in combination with:

$$GoingDown \wedge \neg DOWN \Rightarrow \bigcirc(DOWN \wedge \neg GoingDown) \vee PP(\bigcirc gate, \delta)$$

preserves GRC safety in sparse space-time.

All the typical non-Byzantine assumptions are carried over as well – trains are not spaced so close that the gate has to be raised and lowered before a car can proceed through the crossing, cars obey the gate signals, etc. Furthermore, the distance P could be checked against the gate and train speeds to ensure appropriate safety in that the gate must go from UP to DOWN in no more time than it takes for a train to travel distance P. The minimum and maximum values of train speed are denoted by TS_{min} and TS_{max}, respectively. To verify that the gate has enough time to be lowered between the detection of entry into P, the shortest time for the train to traverse the distance P must be greater than the longest time it takes the gate to be completely raised. In other words:

$$\frac{P}{TS_{max}} > \frac{1}{4 \times GS_{min}}$$

where TS_{max} is in m/s, P is in m, and GS_{min} is in rev/s.

4.2 GRC Liveness

The liveness restriction prevents the gate from being down when no trains are present. This is expressed in spatial-temporal logic as:

$$DC(r, R) \Rightarrow \neg DOWN.$$

Satisfying liveness requires the gate and train dynamics as presented above in the discussion of safety:

$$GoingUp \wedge \neg UP \Rightarrow \bigcirc(UP \wedge \neg GoingUp) \vee PP(\bigcirc gate, \gamma)$$

with the initiating condition:

$$\neg DC(r,R) \wedge \bigcirc DC(r,R) \Rightarrow \bigcirc GoingUp.$$

These fulfill the liveness property with the condition that one additional system state capture may be required when disconnect is detected but the gate has not begun ascending. In other words, detection (or prediction) of $DC(r,R)$ is a prerequisite to beginning the $GoingUp$ sequence of events. Thus, at the system state capture where $DC(r,R)$ is first satisfied, $DOWN$ may also hold. However, at the next system state capture (per $GoingUp$), $\neg DOWN$ will hold. This nuance in the system dynamics is one area where specifications can become bogged down in the details of the language instead of the "big picture" aspects of the specification. However, the combination of such nuances (e.g., sampling frequency, sampling errors and measurement variations) combine to produce formidable challenges to understanding the limits and capabilities of real-world implementations of runtime assertion checking.

5. Comparison of Specifications

Given the GRC example described above, the correct system operation can be discussed in two main ways: (i) an axiomatic specification that prescribes the allowed behavior; and (ii) an operational specification that describes system operation conforming to the allowable behavior. The axiomatic specification comprises invariants that hold over correct system performance. The operational specification provides the mechanistic requirements that produce the desired behavior. Our analysis focuses on the operational specifications of generalized safety and liveness properties, including factors that may impact the human understanding of assertions in each specification instance. The specifications are compared based on the number of required initial and supporting definitions, number of equations and depth of expressions enumerated as a single count of tokens (e.g., comparisons and calculations). These metrics were selected to represent the amount of estimated effort to understand and generate subsequent assertions within the languages.

5.1 Original GRC Specification

The original GRC paper by Heitmeyer and Lynch [6] examined variations of a railroad crossing example and rebuilt the problem from the ground up. Their approach focuses on an axiomatic specification of the safety and liveness properties along with operational specifications of the trains and gates, incorporating timed automata, invariants and simulation mappings to model and verify correct system behavior. Reproducing even a portion of their work here for illustrative purposes is unreasonable due to the iterative nature of their formal methods and the amount of material that is required to show correctness. However, a hint of their process is included to illustrate the comparative metrics.

Table 3. Automaton excerpt.

State	**Transitions**
now	$enterR(r)$
for each train r:	Precondition:
$r.status$	$s.r.status = not\_here$
$first(enterI(r))$	Effect:
$last(enterI(r))$	$s'.r.status = P$
	$s'.first(enterI(r)) = now + \epsilon_1$
	$s'.last(enterI(r)) = now + \epsilon_2$

The Heitmeyer-Lynch axiomatic specification, represented as timed automata, is an initial step in formal verification and helps provide comparative metrics. In the specification, system variables include the upper and lower timing bounds (e.g., time to raise the gate); definitions include listed restrictions such as a lower bound system variable is less than or equal to the corresponding upper bound; and equations, which are transition preconditions or effects in the automaton and safety and liveness specifications. Tokens do not have an exact counterpart in the other GRC specifications, but are obtained from the states and transitions where each comparison or calculation is a token.

As an example, consider the automaton excerpt in Table 3. The excerpt has the system variables ϵ_1 and ϵ_2 (defined previously), four equations (all under transitions) and fourteen tokens. Note that each state, equation operation and reference is considered to be a token. Thus, now and $r.status$ are both calculated as a single token whereas $first(enterI(r))$ and $last(enterI(r))$ are both counted as two tokens because they reference $enterI(r)$.

Table 4. Heitmeyer-Lynch GRC specification.

System Variables	8
Definitions	4
Equations	30
Tokens	89

For the purpose of counting tokens, it is assumed that there is only one train r under the "for each train" in state. In the case of the token counts corresponding to the remaining transitions, the precondition is a single token, the first effect is a single token and the final two effects are each three tokens (containing two equation operations and one reference). Performing these computations over their entire specification yields the results shown in Table 4.

5.2 Shape Calculus GRC Specification

The second model used for comparison is Quesel and Schafer's shape calculus [10]. This specification incorporates discrete time and space, allowing for finite space and infinite time (these choices are due to decidability issues that allow model checking). The discretization corresponds to the numeric discretization of the railroad shown in Figure 2. Key aspects of the shape calculus are included here to assist in understanding the notation and how comparative metrics relate to the GRC specifications.

An observation of a train at a specific point in time and space is *train*; if this holds at all points in time and space (in the interval of interest), then it is expressed as $\lceil \text{train} \rceil$. The "chop" operator, where $\mathcal{F}\langle e_d \rangle \mathcal{G}$ reads $\mathcal{F}$ chop $\mathcal{G}$, specifies that there is a chop point in dimension d (time and/or space) at which $\mathcal{F}$ holds up to and including the chop point, and $\mathcal{G}$ holds at and after the chop point. The diameter of the spatial or temporal dimension d is ℓ_{e_d} and an empty observation interval $\lceil \rceil_{e_d}$ has a diameter of zero. The "somewhere" operator $\Diamond_{e_d}\mathcal{F}$ is defined if and only if $\mathcal{F}$ is true in some subinterval:

$$\Diamond_{e_d}\mathcal{F} \equiv true\langle e_d \rangle \mathcal{F} \langle e_d \rangle true.$$

The globally operator must hold in all subintervals, expressed as the dual:

$$\Box_{e_d}\mathcal{F} \equiv \neg\Diamond_{e_d}\neg\mathcal{F}.$$

This basic syntax can be used to express the shape calculus propositions corresponding to the GRC.

The most basic proposition determines if there is a train within an interval. Using the definitions in [10], this is defined as trainPartWeak:

$$\text{trainPartWeak} \equiv \neg(\lceil \neg \text{train} \rceil \langle e_x \rangle true).$$

However, this must be strengthened to exclude the empty observation interval, so:

$$\text{trainPart} \equiv \text{trainPartWeak} \wedge \ \ell_{e_x} > 0.$$

Or,

$$\text{trainPart} = \neg(\lceil \neg \text{train} \rceil \langle e_x \rangle true) \wedge \ \ell_{e_x} > 0.$$

A distance operator dist(δ) is defined in [10] as:

$$\text{dist}(\delta) \equiv ((\lceil \neg \text{train} \rceil \vee \lceil \rceil_{e_x}) \wedge \ell_{e_x} = \delta) \ \langle e_x \rangle \text{trainPart}.$$

This divides the track so that the rightmost part contains no train and the leftmost part contains the train, providing a measure of distance δ from the train to the end of the observation interval. Expanding this based on first principles yields:

$$\text{dist}(\delta) \equiv ((\lceil \neg \text{train} \rceil \vee \lceil \rceil_{e_x}) \wedge \ell_{e_x} = \delta) \langle e_x \rangle (\neg(\lceil \neg \text{train} \rceil \langle e_x \rangle true) \wedge \ \ell_{e_x} > 0).$$

A careful observation of the propositions shows that $\mathcal{F}$ holds $\lceil\neg$ train$\rceil$ and $\mathcal{G}$ holds $\neg\lceil\neg$ train$\rceil$ (or $\neg\mathcal{F}$). However, at the chop point, both $\mathcal{F}$ and $\mathcal{G}$ must hold, which requires $\mathcal{F} = \neg\mathcal{F}$. Ignoring this technicality and assuming that the shape calculus distance operator holds, the regions of the track can be specified as empty $\equiv \lceil\neg$ train$\rceil$, appr $\equiv dist(\delta) \wedge 10 > \delta \geq 2$, and cross $\equiv dist\delta \wedge 2 > \delta$. Thus, relating these back to the regions in Figure 2, "empty" indicates that no train is present in the region of interest R, "appr" indicates that there is a train in P (approaching) and "cross" indicates that a train is in the crossing region I.

To maintain the physical properties of the system such as restricting the train speed to under the maximum limit and preventing the train from stopping in the crossing indefinitely, Quesel and Schafer introduce a runProgress specification defined as:

$$\text{runProgress} \equiv \Box_{e_t}\Box_{e_x}(((l_{e_x} = MAXSPEED\langle e_x\rangle\text{trainPart}) \wedge l_{e_t} = 1) \langle e_t\rangle l_{e_t} = 1) \Rightarrow (l_{e_t} = 1\langle e_t\rangle\text{trainPart}).$$

Expanding according to the first principles, yields:

$$\begin{aligned}\text{runProgress} \equiv \neg(true\langle e_t\rangle(true\langle e_x\rangle\neg(((l_{e_x} = MAXSPEED\langle e_x\rangle \\ \neg(\lceil\neg\text{train}\rceil\langle e_x\rangle true) \wedge \ell_{e_x} > 0) \wedge l_{e_t} = 1)\langle e_t\rangle l_{e_t} = 1)\langle e_x\rangle true)\langle e_t\rangle true) \\ \Rightarrow (l_{e_t} = 1\langle e_t\rangle\neg(\lceil\neg\text{train}\rceil\langle e_x\rangle true) \wedge\ \ell_{e_x} > 0).\end{aligned}$$

Additional specifications are necessary to complete the shape calculus specification of safety and liveness. However, these initial specifications suffice to illustrate the format of GRC specifications and how they are incorporated within the comparisons.

The definitions in the specification include the chop operator $\mathcal{F}\langle e_d\rangle\mathcal{G}$, diameter $\lceil\ \rceil_{e_d}$ and negation $\neg\mathcal{F}$. Each usage constitutes a token along with standard binary operators ($\wedge$, $\vee$, $>$, $<$, $=$, $\Rightarrow$). Thus, the Equation trainPartWeak given by:

$$\neg(\lceil\neg\text{train}\rceil\langle e_x\rangle true)$$

contains four tokens: the encompassing negation, chop operator, diameter and final negation. Note that train and true are base expressions and not evaluated as tokens. Any equation incorporating other expressions automatically brings in the accompanying tokens (as would be the case when expanded according to first principles). Thus, the specification of trainPart:

$$\text{trainPart} \equiv \text{trainPartWeak} \wedge\ \ell_{e_x} > 0$$

contains six tokens: four from trainPartWeak, and one each from the *and* operator and greater-than comparison. It is interesting to note that, in this specification, the two system variables are MaxSpeed and ReactTime, which are both translated back to the GRC example to relate them to the quantitative,

Table 5. Quesel-Schafer GRC specification.

System Variables	2
Definitions	5
Equations	11
Tokens	129

physical spatial distance that the train can travel during a temporal interval. Table 5 shows the results for the Quesel-Schafer GRC specification.

5.3 Sparse Space-Time GRC Specification

Summarizing the discussion above, the sparse space-time GRC specification holds when the following assertions are satisfied:

$$PP(gate, [DOWN, UP]) \tag{2}$$

$$DC(r, P) \wedge \bigcirc \neg DC(r, P) \Rightarrow \bigcirc GoingDown \tag{3}$$

$$GoingDown \wedge \neg DOWN \Rightarrow \bigcirc (DOWN \wedge \neg GoingDown) \vee PP(\bigcirc gate, \delta) \tag{4}$$

$$\neg DC(r, R) \wedge \bigcirc DC(r, R) \Rightarrow \bigcirc GoingUp \tag{5}$$

$$GoingUp \wedge \neg UP \Rightarrow \bigcirc (UP \wedge \neg GoingUp) \vee PP(\bigcirc gate, \gamma) \tag{6}$$

Equation (2) is a safety constraint that binds the gate between the down and up positions. Equations (3) and (4) initiate and facilitate the lowering of the gate upon train entry. Equations (5) and (6) initiate and facilitate the raising of the gate upon train exit. The sparse space-time system variables include *gate* and r, which denote the position of the gate and train, respectively, along with GS_{min}, GS_{max}, *SamplingRate*, and the Boolean variables *GoingUp* and *GoingDown*. The definitions include the spatial regions P, R, δ and γ. Finally, each RCC relation or operator is counted as a token (e.g., $PP(\bigcirc gate, \gamma)$ is two tokens, one for the Proper Part relation and one for the *next* operator).

5.4 Discussion

Table 6 presents a comparison of all the GRC specifications. The comparison indicates that the sparse space-time approach is more terse than the other specifications based on the system variables, definitions and equations, and counting the number of tokens required to express safety and liveness.

Assertion checking is based on a logical specification of correct operating behavior. It is anticipated that properties of critical infrastructure systems can be incorporated within assertion checking as additional, inherent constraints, which is how expressions can capture system dynamics natively within sparse

Table 6. Comparison of GRC specifications.

GRC Specification	Vars.	Defns.	Eqns.	Tokens
Heitmeyer-Lynch	8	4	30	89
Quesel-Schafer	2	5	11	129
Sparse Space-Time	7	4	5	32

space-time. The result is that assertions can be much more honed to the desired properties (e.g., safety and liveness) and do not have to be explicitly crafted to capture or convert physical properties such as the speeds of gates and trains in the GRC example.

6. Conclusions

The sparse space-time paradigm can be used to naturally express spatial-temporal assertions pertaining to critical infrastructure systems. The utility of the paradigm is demonstrated via a specification of a one-dimensional railway crossing problem, which includes safety and liveness properties. The generalized metrics indicate that, although they may be more terse, sparse space-time assertions are actually simpler to understand and create.

Our future research will explore the application of the sparse space-time paradigm in domains that span multiple dimensions and/or include the modeling of cyber-physical systems. Additionally, we will consider space-time trajectories as fundamental components of the spatial-temporal *next* operator; this will help reduce the dependence on external motion specifications, such as those used for gate dynamics. Our future research will also examine the application of the flow tree concept [13] in conjunction with the sparse space-time paradigm.

Acknowledgements

This research was supported in part by the Future Renewable Electric Energy Distribution Management Center, an NSF Engineering Research Center, under Grant No. EEC 0812121; and by the Missouri S&T Intelligent Systems Center.

References

[1] J. Allen, Maintaining knowledge about temporal intervals, *Communications of the ACM*, vol. 26(11), pp. 832–843, 1983.

[2] B. Clarke, A calculus of individuals based on "connection," *Notre Dame Journal of Formal Logic*, vol. 22(3), pp. 204–218, 1981.

[3] A. Gerevini and B. Nebel, Qualitative spatio-temporal reasoning with RCC-8 and Allen's interval calculus: Computational complexity, *Proceedings of the Fifteenth European Conference on Artificial Intelligence*, pp. 312–316, 2002.

[4] M. Gosnell and B. McMillin, Sparse Space-time Specification of the Generalized Railroad Crossing, Technical Report CSC-12-01, Department of Computer Science, Missouri University of Science and Technology, Rolla, Missouri, 2012.

[5] A. Haxthausen and J. Peleska, Formal development and verification of a distributed railway control system, *IEEE Transactions on Software Engineering*, vol. 26(8), pp. 687–701, 2000.

[6] C. Heitmeyer and N. Lynch, The generalized railroad crossing: A case study in formal verification of real-time systems, *Proceedings of the Real-Time Systems Symposium*, pp. 120–131, 1994.

[7] R. Kontchakov, A. Kurucz, F. Wolter and M. Zakharyaschev, Spatial logic + temporal logic = ? in *Handbook of Spatial Logics*, M. Aiello, I. Pratt-Hartmann and J. van Benthem (Eds.), Springer, Dordrecht, The Netherlands, pp. 497–564, 2007.

[8] H. Kopetz, Sparse time versus dense time in distributed real-time systems, *Proceedings of the Twelfth International Conference on Distributed Computing Systems*, pp. 460–467, 1992.

[9] N. Malik, J. Baumgartner, S. Roberts and R. Dobson, A toolset for assisted formal verification, *Proceedings of the IEEE International Conference on Performance, Computing and Communications*, pp. 489–492, 1999.

[10] J. Quesel and A. Schafer, Spatio-temporal model checking for mobile real-time systems, *Proceedings of the Third International Colloquium on the Theoretical Aspects of Computing*, pp. 347–361, 2006.

[11] D. Randell, Z. Cui and A. Cohn, A spatial logic based on regions and connection, *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, pp. 165–176, 1992.

[12] F. Wolter and M. Zakharyaschev, Qualitative spatiotemporal representation and reasoning: A computational perspective, in *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel (Eds.), Morgan Kaufmann, San Francisco, California, pp. 175–216, 2002.

[13] J. Wood, J. Dykes, A. Slingsby and R. Radburn, Flow trees for exploring spatial trajectories, *Proceedings of the Seventeenth Annual Conference on GIS Research*, pp. 229–234, 2009.

Chapter 15

A NETWORKED EVIDENCE THEORY FRAMEWORK FOR CRITICAL INFRASTRUCTURE MODELING

Chiara Foglietta, Andrea Gasparri, and Stefano Panzieri

Abstract This paper describes a distributed approach for data fusion and information sharing based on evidence theory and the transferable belief model. Evidence theory aggregates data generated from different sources in order to better assess an ongoing situation and to aid in the response and decision making processes. In the domain of critical infrastructure protection, researchers are forced to develop distributed approaches for modeling and control with a minimal exchange of data due to the existence of multiple stakeholders and interconnections between infrastructure components. Evidence theory permits the modeling of uncertainty in data fusion, but it is typically applied in a centralized manner. This paper proposes a decentralized extension of the transferable belief model that facilitates the application of evidence theory to data fusion in critical infrastructure applications. A case study is provided to demonstrate the convergence of results similar to the centralized approach, and to show the utility of fusing data in a distributed manner for interdependent critical infrastructure systems.

Keywords: Modeling, evidence theory, situational awareness, data fusion

1. Introduction

A nation's critical infrastructure comprises complex, interdependent systems whose proper operation and interaction are essential to the welfare of society. Modeling the interdependencies between the different critical infrastructure sectors is a complex task, but this is vital to correctly analyze and predict cascading phenomena.

Interdependencies, from the viewpoint of data fusion, can be analyzed to discern critical events and failures. Events associated with critical infrastructures typically have low probabilities but high impact. The impact is exacerbated

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 205–215, 2012.

by effects that are not localized and tend to propagate to interconnected infrastructures. This implies that a "signature" of the event can be identified in the interconnected system and, sometimes, in independent agencies such as meteorological services or police departments. Therefore, there is a need to fuse together the available data and derive a common belief of the current situation. Developing a common belief facilitates efficient and effective decisions and plans of action. Since critical infrastructures are inherently distributed [5], the fusion mechanism should also be distributed.

This paper provides an extension of the transferable belief model to facilitate the application of evidence theory. The existing transferable belief model provides a centralized technique for fusing data. However, a decentralized approach is required for interdependent critical infrastructures. This paper demonstrates the application of an extended decentralized transferable belief model to the domain of critical infrastructure protection and shows how data fusion can help develop more accurate beliefs about interdependent infrastructures.

2. Background

Evidence theory is a methodology that is commonly applied to data fusion problems. Data fusion seeks to combine data from heterogeneous sources or sensors to provide estimates of ongoing events [3].

Evidence theory stems primarily from the pioneering work of Dempster [4] and Shafer [10] and is often considered in the same light as Bayesian networks. The primary difference, however, is the ability to deal with uncertainty [9]. The Dempster-Shafer theory [10] explicitly considers uncertainty and examines if the various sources of data are inconsistent or if there is an error in the modeling process. On the other hand, Bayesian networks use the recognition of input values as a likelihood from pre-determined patterns. The application of the two approaches depends on the type of knowledge that has to be represented and fused [8].

Several methods have been proposed for combining and correlating the available data in the context of the Dempster-Shafer framework. This work uses the methodology proposed by Smets [11], which extends the Dempster-Shafer framework by assuming that the correct answer might not be among the considered ones (i.e., it engages the open world assumption). Smets' approach also allows the computation of the amount of contradictory data in the value of the empty set.

The major limitation to applying evidence theory in a real context is the number of hypotheses required to model the application of interest. This can be explained by the fact that, from a computational perspective, the power set of the set of hypotheses has to be computed, causing the complexity to grow exponentially with the number of hypotheses. However, evidence theory has been successfully applied to a variety of practical problems using approximations (see, e.g., [7, 13]).

Data fusion can also aid in impact assessment. In fact, limiting an impact assessment strictly to measured events can lead to heavy underestimation or in-

correct estimation. For example, an isolated failure can propagate in a number of ways depending on the cause and the detection methods. Examples include a fire blast detected by a sensor and a computer virus detected by an intrusion detection system. In the first example, the fire blast propagation effects are associated with an interdependency model according to a spatial proximity pattern. In the second example, a computer virus may propagate to similar, and highly dispersed, telecommunication nodes.

3. Evidence Theory

Evidence theory provides a means to form a consolidated belief by correlating evidence from different sources [4, 10, 11]. This section provides an overview of the evidence theory formalisms used in this work.

Let ω_i represent a cause of system failure and $\Omega = \{\omega_1, \cdots, \omega_n\}$ be the set of hypotheses containing known possible failures. This set is called the "frame of discernment." For example, the possible causes of failures of a critical infrastructure asset include sabotage, device failure, fault due to weather and a (cyber) denial-of-service attack. Note that the hypotheses are assumed to be mutually exclusive in evidence theory.

The power set of the frame of discernment is expressed as $\Gamma(\Omega) = \{\gamma_1, \cdots, \gamma_{2^{|\Omega|}}\}$, which has cardinality $|\Gamma(\Omega)| = 2^{|\Omega|}$. The power set contains all possible subsets of Ω, including the empty set $\gamma_1 = \emptyset$ and the universal set $\gamma_{2^{|\Omega|}} = \Omega$.

The transferable belief model [11] is derived from the basic belief mass function m:

$$m : \Gamma(\Omega) \rightarrow [0, 1].$$

This function, also called the "basic belief assignment" (BBA), maps each element of the power set to a value between 0 and 1. Each BBA is an atomic element in the transferable belief model. In fact, each sensor, agent or node must be able to assign the BBA values by some subjective assumptions or through algorithms that automatically determine the assignment. The BBA function is constrained by:

$$\sum_{\gamma_a \subseteq \Gamma(\Omega)} m(\gamma_a) = 1 \quad \text{with} \quad m(\emptyset) = 0.$$

The transferable belief model examines propositions accordingly as: "the true value of ω_i is in γ_a" where $\gamma_a \in \Gamma(\Omega)$. For $\gamma_a \in \Gamma(\Omega)$, $m(\gamma_a)$ is the confidence that supports exactly γ_a. This implies that the true value is in the set γ_a; however, due to the lack of additional data, it is not possible to support any strict subset of γ_a.

In the case of different independent data sources, a rule is necessary to aggregate the data. Several rules of combination exist in the literature; the most widely used rules are Dempster's rule [4] and Smets' rule [11].

Dempster's rule of combination [4], which was the first to be proposed, is a purely conjunctive operation. The rule strongly emphasizes the agreement between multiple sources and ignores conflicting evidence using a normalization

factor as shown in the following equation:

$$\text{Dempster}\{m_i, m_j\}(\emptyset) = 0$$

$$\text{Dempster}\{m_i, m_j\}(\gamma_a) = \frac{\sum_{\gamma_b \cap \gamma_c = \gamma_a} m_i(\gamma_b) m_j(\gamma_c)}{1 - \sum_{\gamma_b \cap \gamma_c = \emptyset} m_i(\gamma_b) m_j(\gamma_c)} \qquad \forall \gamma_a \in \Gamma(\Omega).$$

On the other hand, Smets' rule of combination [11] provides the ability to explicitly express the contradiction in the transferable belief model by letting $m(\emptyset) \neq 0$. This combination rule, unlike Dempster's rule, avoids normalization while preserving commutativity and associativity:

$$\text{Smets}\{m_i, m_j\}(\gamma_a) = m_i(\gamma_a) \otimes m_j(\gamma_a) \qquad \forall \gamma_a \in \Gamma(\Omega)$$

where

$$m_i(\gamma_a) \otimes m_j(\gamma_a) = \sum_{\gamma_b \cap \gamma_c = \gamma_a} m_i(\gamma_b) m_j(\gamma_c) \qquad \forall \gamma_a \in \Gamma(\Omega).$$

The relation $m(\emptyset) > 0$ can be explained in two ways: (i) open world assumption; and (ii) quantified conflict. The open world assumption, proposed by Dempster, reflects the idea that the frame of discernment must contain the true value. If the open world assumption is true, then the set of hypotheses must contain all possibilities. Under this interpretation, if $\emptyset$ is the complement of Ω, the mass $m(\emptyset) > 0$ represents the case where the truth is not contained in Ω. Alternatively, the notion of quantified conflict means that there is some underlying conflict between the sources that are combined to produce the BBA. Hence, the mass assigned to $m(\emptyset)$ represents the degree of conflict. Specifically, it is computed as:

$$m_i(\emptyset) \otimes m_j(\emptyset) = 1 \quad - \sum_{\gamma_a \in \Gamma, \gamma_a \neq \emptyset} \left(m_i(\gamma_a) \otimes m_j(\gamma_a) \right).$$

4. Data Fusion in the Network Context

Consider a network of multiple agents described by an indirect graph $\mathcal{G} = \{V, E\}$ where $V = \{v_i \mid i = 1, \cdots, n_V\}$ is the set of nodes and $E = \{e_{ij} \mid (v_i, v_j)\}$ is the set of edges that represent a communication channel between the nodes. Edges are indirect and, thus, the existence of an arc e_{ij} implies the existence of an edge e_{ji}.

In this work, we assume that no central unit is available to perform data aggregation. Additionally, communications between nodes are limited to the neighbors of the node under consideration (i.e., nodes that are physically or directly connected to the node under consideration). These assumptions are reasonable for data fusion problems in the area of sensor networks.

Table 1. BBA assignments for a telecommunications network.

Set	Node 1	Node 2	Node 3	$\mathbf{m_{12}}$	$\mathbf{m_{123}}$
$\emptyset$	0.0	0.0	0.0	0.44	0.770
{a}	0.1	0.5	0.7	0.11	0.095
{b}	0.8	0.4	0.2	0.44	0.134
{a,b}	0.1	0.1	0.1	0.01	0.010

A direct consequence of the assumptions, however, is that Smets' rule of composition cannot be directly applied. Indeed, applying Smets' rule multiple times over the same BBAs leads to different outputs. Note that this could easily happen in a distributed context where communications are local and limited to the one-hop neighborhood.

Consider, for example, a scenario where it is necessary to identify the degradation of services in a telecommunications network. In the case of extensive delays in packet transmission, it may be necessary to determine if the situation is a temporary congestion or a persistent malicious attack. Network sensors (e.g., intrusion detection systems) can detect a denial-of-service (DoS) attack. The DoS attack may result in cascading effects within the network that are identified by other sensors. If the attack is conducted on a sufficiently large scale, entire regions can be compromised, with different sensors providing different inputs based on localized knowledge.

A general method to define a BBA allocation is to consider the reliability of the data source. Suppose that the data obtained from the source supports a set of hypotheses in Ω. Then, the subset γ_a of the power set $\Gamma(\Omega)$, containing the set of hypotheses, receives a mass $m(\gamma_a)$ equal to the reliability of the source. The remaining mass $1 - m(\gamma_a)$ is assigned to the universal set because no other data is available.

Table 1 shows the BBA assignments for a cause classification problem in a telecommunications network using the centralized approach. The network has three sources (nodes) that relay data to determine the probable causes of a network congestion incident. The table shows the BBA values assigned to the various network nodes. The frame of discernment is $\Omega = \{a, b\}$, where hypothesis a is a denial-of-service attack (i.e., congestion of one or more network nodes that intentionally degrades the telecommunications network), and hypothesis b indicates congestion in the telecommunications network due to routing problems.

If we assume that Node 1 is coordinating with Node 2, then upon applying Smets' operator, the result is:

$$m_1 \otimes m_2 = \{0.44, 0.11, 0.44, 0.01\}.$$

Algorithm 1 : Gossip Algorithm

Data: $t = 0$, $s_i(t = 0) \ \ \forall i = 1, \cdots, n$
Results: $s_i(t_{end}) \ \ \forall i = 1, \cdots, n$

while *end_condition* **do**
Select an edge $e_{ij} \in E(t)$ according to $\mathbf{e}$;
Update the states of the selected nodes by applying the operator $\mathcal{R}$:
$s_i(t+1) = s_i(t) \otimes s_j(t)$
$s_j(t+1) = s_j(t) \otimes s_i(t)$
Set $t = t + 1$
end

If Node 1 then communicates with Node 3 by exchanging data about the possible cause of the congestion, then the result obtained by centralized aggregation is equal to:

$$m_{12} \otimes m_3 = \{0.77, 0.095, 0.134, 0.001\}.$$

Next, we consider the communications between Node 1 and Node 3 by applying Smets' operator. The result, which is different from the one obtained in the centralized system, is given by:

$$m_{123} \otimes m_3 = \{0.8828, 0.0767, 0.0404, 0.0001\}.$$

If a decision on the cause of the fault has to be made, then the latter case results in a change of opinion; according to the latest aggregations, the cause is a denial-of-service attack (hypothesis a) instead of network routing congestion (hypothesis b).

As shown, Smets' rule of combination cannot be directly used in a distributed data aggregation context. The next section presents a distributed algorithm that can update the knowledge of all the nodes in a network.

5. Data Fusion Algorithm

The algorithm proposed by Gasparri, *et al.* [6] provides the ability to divide the knowledge of each node into two parts: (i) data shared between two nodes; and (ii) localized data retained by each node.

In the decentralized algorithm, the network is described by an indirect graph $\mathcal{G} = \{V, E\}$. A spanning tree $\mathcal{T} = \{V, \widehat{E}\}$ is derived, where $\widehat{E} \subseteq E$ is available to all nodes. Note that the nodes must have the capacity to save data. Interested readers are referred to [2] for details about spanning tree construction.

Communications between nodes are asynchronous and follow the Gossip Protocol [1]. This protocol is formalized as Algorithm 1. The triplet $\{\mathcal{S}, \mathcal{R}, \mathbf{e}\}$ specifies the network such that:

- $\mathcal{S}$ is the set of local states of each node in the network.

- $\mathcal{R}$ is local interaction rule, i.e., for each pair of nodes (i, j) such that $e_{ij} \in E$, the following equation holds:

$$\mathcal{R} : \mathbb{R}^q \times \mathbb{R}^q \rightarrow \mathbb{R}^q$$

  where q is the number of elements called "focal sets."

- **e** is the process of edge selection for which $e_{ij} \in E(t)$ is the selected edge at time t.

To define the operator $\mathcal{R}$, it is first necessary to introduce the operator $\odot$. Consider two sets of BBAs:

$$m_k = \{m_k(\gamma_a) \mid \forall \gamma_a \in \Gamma(\Omega)\}$$

and

$$m_i = \{m_i(\gamma_a) \mid \forall \gamma_a \in \Gamma(\Omega)\},$$

such that $m_k = m_i \otimes m_j$. The operator $\odot$ can then be defined as:

$$m_j = m_k \odot m_i \triangleq \tilde{m}_k^i.$$

Starting with the power set element with the highest cardinality and examining elements with decreasing cardinality, the value of a BBA can be computed recursively as follows:

$$m_j(\gamma_a) = \frac{m_k(\gamma_a) \; - \sum\limits_{\gamma_b \cap \gamma_c = \gamma_a, \gamma_a \subset \gamma_b} m_j(\gamma_b) m_i(\gamma_c)}{\sum\limits_{\gamma_a \subseteq \gamma_b} m_i(\gamma_b)}.$$

It is now possible to introduce the operator $\mathcal{R}$, along with $\oplus$, to aggregate BBAs of the nodes:

$$m_i(t+1) = m_j(t+1) = m_i(t) \oplus m_j(t)$$
$$= \{\left(\tilde{m}_i^j(t, \gamma_a) \otimes \tilde{m}_i^j(t, \gamma_a)\right) \otimes \bar{m}_{i,j}(t, \gamma_a), \forall \gamma_a \in \Gamma(\Omega)\}.$$

Note that the term $\tilde{m}_i^j(t, \gamma_a)$ indicates the innovation of node i with respect to the node j, which can be calculated recursively using the operator $\mathcal{S}$:

$$\tilde{m}_i^j(t, \gamma_a) = m_i(t, \gamma_a) \odot \bar{m}_{i,j}(t, \gamma_a).$$

The element $\bar{m}_{i,j}(t, \gamma_a)$ express the common knowledge (i.e., knowledge exchanged between two agents (i, j) after the last aggregation) such that:

$$\bar{m}_{i,j}(t, \gamma_a) = \mathbf{n} = \{0, 0, \cdots, 0, 1\}.$$

Gasparri, *et al.* [6] have shown that the algorithm converges to the same result as the centralized aggregation algorithm. The convergence time of the distributed algorithm is related to the diameter d of the spanning tree. Additionally, the computational complexity of the $\mathcal{R}$-operator is the same as that of Smets' operator.

6. Application Scenario

Applying the transferable belief model involves modeling a problem, specifying the frame of discernment and selecting the BBAs. Note that the assignment of BBAs is problem dependent and depends significantly on the source of information and knowledge about the system. In this work, we assume that experts provide advice on assigning the BBAs. Other possible approaches are described in [3, 12], where the reliability of the data sources is considered along with the effect of mass assignment on the compound hypotheses.

Our scenario involves a supervisory control and data acquisition (SCADA) system that aggregates data from a large number sensors positioned at remote locations. Operators located at a control center manage operations by acquiring system data and updating system parameters. Note that the remote facilities all have data regarding critical events and that the data is generated from different sources.

Communications between the control center and the devices in the field occur via dedicated telecommunications circuits or shared public media such as the Internet. Note that the same communications channels can support information sharing among critical infrastructures to help discern the possible causes of failures. The concept of information sharing across sectors can help prevent cascading effects or prevent the impact from propagating to interconnected infrastructures that have not yet been affected.

Our scenario considers $n = 5$ interdependent critical infrastructures. Each infrastructure owner determines the BBAs in his/her infrastructure. To discern the cause of a fault, the BBAs have to be aggregated and shared among the five infrastructures. The frame of discernment is $\Omega = \{a, b, c\}$, where hypothesis a represents a cyber attack, hypothesis b represents the failure of an isolated single unit, and hypothesis c represents a natural disaster (e.g., earthquake). Table 2 shows the BBA assignments for the five infrastructure nodes.

First, we evaluate the scenario using the centralized algorithm. The results as shown in Table 3. Each column presents the results obtained using Smets' operator, with column "NN 12" representing the aggregation of Nodes 1 and 2, column "NN 123" representing the aggregation of Nodes 1, 2 and 3, and so on. The last column "C-TBM" shows the results for the centralized transferable belief model.

Next, we evaluate the scenario using the distributed algorithm. Table 4 shows the edge selection data. Table 5 shows the output of the $\mathcal{R}$-operator. Each column corresponds to the aggregation of two nodes as related to the sequential timing.

Table 2. BBA assignments for the five nodes.

Set	Node 1	Node 2	Node 3	Node 4	Node 5
∅	0.0	0.0	0.0	0.0	0.0
{a}	0.3	0.0	0.0	0.0	0.2
{b}	0.3	0.4	0.1	0.4	0.4
{c}	0.0	0.4	0.5	0.4	0.1
{a,b}	0.3	0.0	0.0	0.0	0.0
{a,c}	0.0	0.0	0.0	0.0	0.0
{b,c}	0.0	0.0	0.3	0.0	0.0
{a,b,c}	0.1	0.2	0.1	0.2	0.3

Table 3. Centralized algorithm output with incremental aggregation.

Set	NN 12	NN 123	NN 1234	NN 12345	C-TBM
∅	0.36	0.468	0.5304	0.6334	0.6334
{a}	0.18	0.108	0.0648	0.0451	0.0451
{b}	0.34	0.346	0.3676	0.3070	0.3070
{c}	0.04	0.046	0.0308	0.0125	0.0125
{a,b}	0.06	0.024	0.0048	0.0014	0.0014
{a,c}	0.0	0.0	0.0	0.0	0.0
{b,c}	0.0	0.006	0.0012	0.0004	0.0004
{a,b,c}	0.20	0.002	0.0004	0.0001	0.0001

Table 4. Temporal edge selection.

Time	t=1	t=2	t=3	t=4	t=5	t=6	t=7
Edge	e_{12}	e_{23}	e_{34}	e_{45}	e_{34}	e_{23}	e_{12}

After each exchange of knowledge between two nodes, the $\mathcal{R}$-operator reveals that the two nodes have the same knowledge value as the operator output. For example, at $t = 5$, the nodes have the same value:

$$\{0.6334, 0.0451, 0.3070, 0.0125, 0.0014, 0.0, 0.0004, 0.0001\}$$

and are consistent with the centralized transferable belief model outputs. Note that the algorithm terminates when the nature of the edge selection process is known [6]. The results demonstrate that the decentralized approach (Table 5) yields the same values as the centralized approach (Table 3).

7. Conclusions

The decentralized extension of the transferable belief model enables the application of evidence theory to data fusion in critical infrastructure applications.

Table 5. Distributed algorithm output.

Set	$s_1 \oplus s_2$	$s_2 \oplus s_3$	$s_3 \oplus s_4$	$s_4 \oplus s_5$	$s_3 \oplus s_4$	$s_2 \oplus s_3$	$s_1 \oplus s_2$
$\emptyset$	0.36	0.468	0.5304	0.6334	0.6334	0.6334	0.6334
{a}	0.18	0.108	0.0648	0.0451	0.0451	0.0451	0.0451
{b}	0.34	0.346	0.3676	0.3070	0.3070	0.3070	0.3070
{c}	0.04	0.046	0.0308	0.0125	0.0125	0.0125	0.0125
{a,b}	0.06	0.024	0.0048	0.0014	0.0014	0.0014	0.0014
{a,c}	0.0	0.0	0.0	0.0	0.0	0.0	0.0
{b,c}	0.0	0.006	0.0012	0.0004	0.0004	0.0004	0.0004
{a,b,c}	0.20	0.002	0.0004	0.0001	0.0001	0.0001	0.0001

This is important because centralized computation is ill-suited to critical infrastructure applications due to the associated interdependencies. The case study shows that the decentralized approach produces the same results as the centralized approach, and also demonstrates the utility of fusing data in a distributed manner for interdependent critical infrastructures.

Integrating situational awareness with distributed monitoring is an appealing concept for networked critical infrastructures. This is important because the ability to leverage and share sensor data can significantly enhance system resilience and robustness. For example, in the case of a cyber attack, data from intrusion detection systems coupled with data from standard field sensors can be combined to obtain more accurate belief assessments.

The decentralized approach offers the same advantages as the traditional transferable belief model in terms of its ability to deal with uncertainty. It is important to note, however, that the same disadvantages exist (e.g., exponential growth in the computational complexity with respect to the number of hypotheses). Nevertheless, both approaches are useful for modeling beliefs in interdependent infrastructures for the purpose of enhancing situational awareness.

Acknowledgement

This research was partially supported by the European Commission through the FP7 Cockpit CI Project.

References

[1] S. Boyd, A. Ghosh, B. Prabhakar and D. Shah, Randomized gossip algorithms, *IEEE Transactions on Information Theory*, vol. 52(6), pp. 2508–2530, 2006.

[2] Y. Dalal, A distributed algorithm for constructing minimal spanning trees, *IEEE Transactions on Software Engineering*, vol. 13(3), pp. 398–405, 1987.

[3] S. Das, *High-Level Data Fusion*, Artech House, Norwood, Massachusetts, 2008.

[4] A. Dempster, Upper and lower probabilities induced by a multivalued mapping, in *Classic Works of the Dempster-Shafer Theory of Belief Functions, Studies in Fuzziness and Soft Computing*, R. Yager and L. Liu (Eds.), Springer, Berlin-Heidelberg, Germany, pp. 57–72, 2008.

[5] S. De Porcellinis, S. Panzieri and R. Setola, Modeling critical infrastructure via a mixed holistic reductionistic approach, *International Journal of Critical Infrastructures*, vol. 5(1/2), pp. 86–99, 2009.

[6] A. Gasparri, F. Fiorini, M. DiRocco and S. Panzieri, A networked transferable belief model approach for distributed data aggregation, *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 42(2), pp. 391–405, 2012.

[7] J. Hall and J. Lawry, Generation, combination and extension of random set approximations to coherent lower and upper probabilities, *Reliability Engineering and System Safety*, vol. 85(1-3), pp. 89–101, 2004.

[8] F. Jensen and T. Nielsen, *Bayesian Networks and Decision Graphs*, Springer, New York, 2007.

[9] K. Ng and B. Abramson, Uncertainty management in expert systems, *IEEE Expert*, vol. 5(2), pp. 29–48, 1990.

[10] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, New Jersey, 1976.

[11] P. Smets and R. Kennes, The transferable belief network, *Artificial Intelligence*, vol. 66(2), pp. 191–234, 1994.

[12] A. Veremme, D. Dupont, E. Lefevre and D. Mercier, Belief assignment on compound hypotheses within the framework of the transferable belief model, *Proceedings of the Twelfth International Conference on Information Fusion*, pp. 498–505, 2009.

[13] F. Voorbraak, A computationally efficient approximation of Dempster-Shafer Theory, *International Journal of Man-Machine Studies*, vol. 30(5), pp. 525–536, 1989.

Chapter 16

ENABLING THE EXPLORATION OF OPERATING PROCEDURES IN CRITICAL INFRASTRUCTURES

Christos Siaterlis, Bela Genge, Marc Hohenadel, and Marco Del Pra

Abstract Modern testbeds for the experimental analysis of critical infrastructures either totally ignore the human factor or incorporate real human-machine interfaces (HMIs) and software that require the presence of human operators during an experiment. Although experimentation with humans in the loop can provide invaluable experimental data about human decision making and reactions, it is infeasible to conduct a systematic exploration of the vast parameter space of possible human operator decisions, reasoning and actions. This paper argues that testbeds should incorporate simulated human decision making capabilities in order to engage humans in the loop, especially because humans play crucial roles in cyber security experiments involving critical infrastructures. An extension of a previously developed experimentation framework is also described; the extension provides generic "human decision" units that enable the integration of human operator and HMI models. The utility of the approach is demonstrated by assessing the impact of human operator reactions during an attack on a cyber-physical infrastructure incorporating the IEEE 30-bus power grid model.

Keywords: Critical infrastructures, operating procedures, security, simulation

1. Introduction

Most investigations that focus on critical infrastructures (e.g., power plants, smart grids and water treatment facilities) highlight the fact that human operators play a crucial role in the resolution of cyber security incidents [7]. Simple configuration mistakes that leave systems unprotected are often uncovered only after security incidents. On the other hand, human decisions can make the difference between a complete breakdown and system survival.

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 217–233, 2012.

The interaction of human operators with critical infrastructures is mostly implemented using information and communications technologies, largely due to reduced costs and greater efficiency, flexibility and interoperability. Consequently, several approaches have focused on the design of (graphical) interfaces, also known as human-machine interfaces (HMIs), that assist decision making processes and reduce the reaction times of human operators [12, 18]. On the other hand, human operators can also interact with a system independently of HMIs, for example, by switching a device on or off. Therefore, testbeds that focus on the analysis of critical infrastructures should also take into account the presence of human operators.

This paper argues that the presence of human operators and HMIs dramatically changes system behavior and should be taken into account when designing testbeds for analyzing critical infrastructures. Existing testbeds (see, e.g., [5, 11, 13, 21]) may engage human operators and real HMIs, but they do not include software simulations of these components. Although testbeds with human operators and real HMIs provide reliable experimental data, they are unable to support exhaustive parameter testing. This is mainly due to the costs involved in acquiring and training human operators, and the costs of customizing proprietary HMI software.

Recognizing the importance of human operators and, especially, operating procedures in security experiments, we propose an extension to our previously developed experimentation framework [6]. The extension incorporates "human decision" units that can run human operator and HMI models in real time. Actions issued by the models are translated into commands that are executed in the cyber and physical realms. This way, the extended framework enables the recreation of realistic scenarios in which operators interact with the cyber realm and also execute actions in the physical realm. The approach is evaluated by assessing the impact of human operator reactions during an attack on a cyber-physical infrastructure incorporating the IEEE 30-bus power grid model.

2. Related Work

Most of the critical infrastructure experimentation testbeds in use today do not take into account human operator or HMI models. In contrast, several testbeds have been developed that model the interactions of human operators with a physical process through real HMIs. The most relevant approaches from both these categories are discussed in this section.

Chabukswar, *et al.* [3] used the Command and Control WindTunnel [14] multi-model simulation environment based on the IEEE high-level architecture standard [19] to facilitate interactions between simulation engines. They used OMNeT++ to simulate a network, and Matlab Simulink to build and run the physical process model. In this approach, neither the human nor the HMI were considered because the main focus of the testbed was to recreate critical infrastructures and hardware control loops.

Hopkinson, *et al.* [9] adopted a similar approach, in which a PowerWorld server (a high-voltage power system simulation and analysis package) [15], was

used to provide a simulation environment for electrical power systems and the ns-2 network simulator was used to simulate other system components (e.g., programmable logic controllers (PLCs) and malware). Like the work of Chabukswar, *et al.*, this approach also does not consider human operators or HMI models. Nevertheless, it provides a generic interface for extending the testbed with additional ns-2 modules. Although the approach could support the integration of external modules, it was not designed to run complex mathematical models such as those developed using dedicated modeling tools like Simulink. Furthermore, designers would have to define "glue" code to enable the integration of a wide range of models with ns-2. These aspects are the main focus of the work described in this paper – among other things, glue code consisting of several modules and interfaces is proposed to integrate human operator and HMI models in a previously developed critical infrastructure experimentation testbed.

In contrast with the two approaches mentioned above, several testbeds (e.g., [5, 11, 13, 21]) incorporate "real" humans and HMIs – specifically, human operators interact with real proprietary HMIs. Although such testbeds provide reliable experimental data, they are unable to support exhaustive parameter testing involving human operators and HMIs. Queiroz, *et al.* [16] have implemented just such an approach. In their testbed, HMIs are simulated as OMNeT++ modules and human operators interact with the simulated HMIs. Although this approach represents an advancement over the approaches described above, it still requires human operators to be present and interact with the system. Ultimately, the fidelity of such an approach is counterbalanced by its higher costs and lower efficiency because experiments might require the presence of multiple humans to perform repetitive tasks in order to cover the entire parameter space.

3. Problem Statement

In many fields, there is an increasing trend to replace humans with automated control loops. Nevertheless, human operators continue to play a significant role in the operation of critical infrastructures, especially during abnormal situations and contingencies.

Most critical infrastructure experimentation testbeds available today (see, e.g., [5, 11, 13, 21]) engage real human operators and HMIs, not simulations of human operators and HMIs. Human operator modeling (see, e.g., [1, 4, 8, 10, 17, 23]) is a complex task; in fact, research on designing human operator models has been around since the beginning of the 20th century [1, 4]. Recent research has demonstrated the applicability of linear and nonlinear control theories [10] and belief-desire-intention paradigms combined with agent-based platforms [17, 23] in the human operator modeling process. Each approach comes with its own advantages and disadvantages, complicating the task of selecting an approach. Therefore, the design of a generic experimentation platform with human operator and HMI models that could be applied to a wide range of critical infrastructures is not a trivial task.

Taking into account the complexity and diversity of critical infrastructures and the operator models needed for each of these systems, a single operator model that is applicable to all systems may not be possible, let alone feasible. A solution that would require the integration of every operator model from scratch would also be infeasible. A more reasonable solution would provide designers with several interfaces that allow a wide variety of operator models to be adapted, coupled and integrated into a testbed. The solution should also enable the coupling and integration of various HMI models because HMI software has a major impact on the state of a critical infrastructure.

It is also important to take into account the fact that large critical infrastructures have many human operators who supervise and control systems at any given time. A generic representation of the human decision making process should include hierarchical and graph-based information flows. These complex interactions should not be ignored or the applicability of the approach would be greatly limited.

Another important issue to be considered is the translation of simulated operator decisions to actions in the cyber-physical realm. These actions include interactions with the physical process as well as interactions in the cyber realm (e.g., configuring a firewall, launching an external script and shutting down a computer). The translation process should be flexible and should rely on generic modules that are easily replaceable. In this way, the implemented solution would support experimentation with a wide range of physical processes and networked industrial control systems.

Extending existing critical infrastructure experimentation testbeds with human operator and HMI models is a complex task. But it is important because, when human operator and HMI models are added, the experimentation environment is able to support the human-in-the-loop paradigm that plays a crucial role in the outcome of any cyber security experiment involving critical infrastructures.

4. Proposed Approach

This section presents our approach for integrating human decision making into our previously developed critical infrastructure experimentation framework [6]. The section begins with a brief overview of the experimentation framework and proceeds to describe the extension.

4.1 Experimentation Framework

The experimentation framework developed in our previous work [6] follows a hybrid approach, where the Emulab-based testbed recreates the control and process network of a networked industrial control system, including PLCs and SCADA servers, and a software simulation reproduces the industrial process. The architecture shown in Figure 1 has three layers: (i) cyber layer; (ii) physical layer; and (iii) link layer, which lies in between the cyber and physical layers. The cyber layer includes various information and communication technology

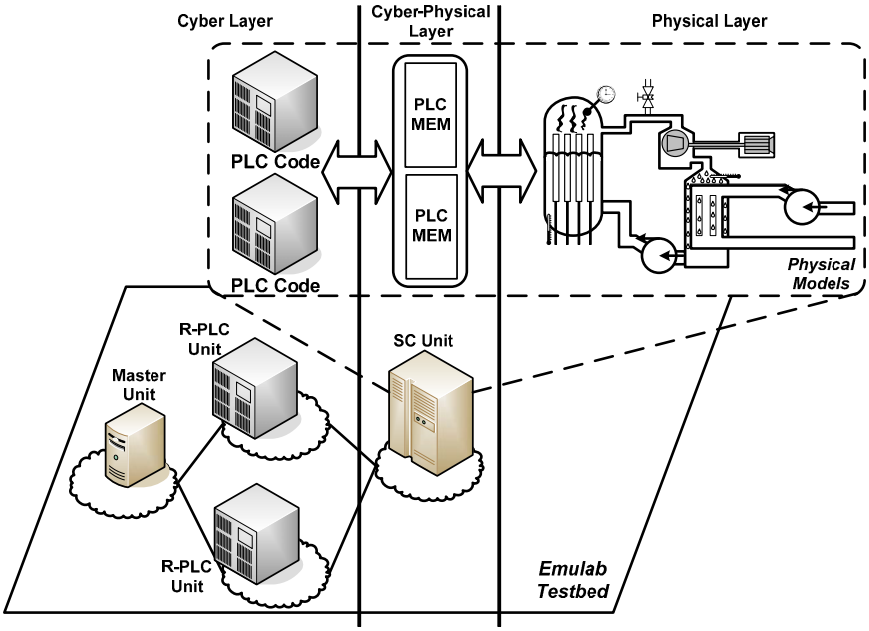


Figure 1. Experimentation framework.

components that are used in SCADA systems, while the physical layer incorporates simulations of physical processes and devices. The link layer (cyber-physical layer) provides the glue between the two layers through the use of a shared memory region.

The physical layer is recreated using a soft real-time simulator that runs on the simulation core (SC unit) and executes a model of the industrial process. The cyber layer is recreated by an emulation testbed that uses the Emulab architecture and software [22] to automatically and dynamically map physical components (e.g., servers and switches) to a virtual topology.

In addition to the process network, the cyber layer also includes control logic code that is executed by PLCs in a real-world system. The control code can be run sequentially or in parallel with the physical model. In the sequential case, tightly coupled code (TCC) is used; this code runs on the SC unit in the same memory space as the model. In the parallel case, loosely coupled code (LCC) is used; this code runs in another address space, possibly on another host in the remote PLC (R-PLC) unit. The main advantage of TCC is that it does not miss values generated by the model between executions. On the other hand, LCC allows the remote execution of PLC code, the injection of malicious code without stopping model execution, and the execution of complex PLC emulators. A master unit implements global decision algorithms based on the sensor values received from R-PLC units.

The cyber-physical layer incorporates PLC memory (as a set of registers typical of PLCs) and the communications interfaces that glue the other two layers. Memory registers provide links to inputs (e.g., valve position) and outputs (e.g., sensor values) of the physical model.

Prototypes of SC, R-PLC and master units were written in C# (Windows) code, which was ported and tested on Unix-based systems (FreeBSD, Fedora and Ubuntu) using the Mono platform. Matlab Simulink is used as the industrial process simulator (physical layer). Matlab Real Time Workshop is used to generate C code from the Simulink models. Communications between SC and R-PLC units are handled by a .NET binary implementation of RPC over TCP (referred to as "remoting"). Communications between the master and R-PLC units use the Modbus TCP protocol.

4.2 Extended Architecture

As mentioned above, adding human decision making to an experimentation testbed offers several advantages. However, the complexity of modeling human decision making requires an approach that does not limit the testbed to one specific model. Therefore, we propose a generic approach for integrating models in cyber-physical testbeds. In this approach, models are seen as "black boxes" that must implement a standard interface in order to interact with other system components. The required interface includes a set of inputs and a set of outputs that are connected to "action scripts" at run time. All the actions issued or received by models are sent through action scripts that include the code necessary for processing actions and communicating with other software components.

To implement the proposed approach, we extended the framework developed in our previous work with a generic human decision (HD) unit. In the remainder of this section, we describe the prototype implementation in detail and provide insight into some of the key aspects of the implementation.

The design of the HD unit started with the assumption that human operators interact with cyber-physical systems in several ways. Human operators certainly rely on information and communications hardware and software present in networked industrial control system installations (e.g., HMIs). However, they also interact independently of these components via installation-specific components (e.g., customized script for configuring a firewall) and physical actions (e.g., shutting down a server). The HD unit architecture takes all these aspects into account through the actions (glue) layer and through action scripts that implement the specifics of each experiment conducted using the testbed.

Figure 2 presents the extended architecture, which includes an HD unit and other components typical of networked industrial control systems. Within the HD unit, human operator and HMI models interact with the cyber-physical realm through an action module that enables bidirectional communications and command execution (i.e., actions). The human operator and HMI models shown in Figure 2 interact in real time using direct connections that are included in the architecture. Although the human operator and HMI models are con-

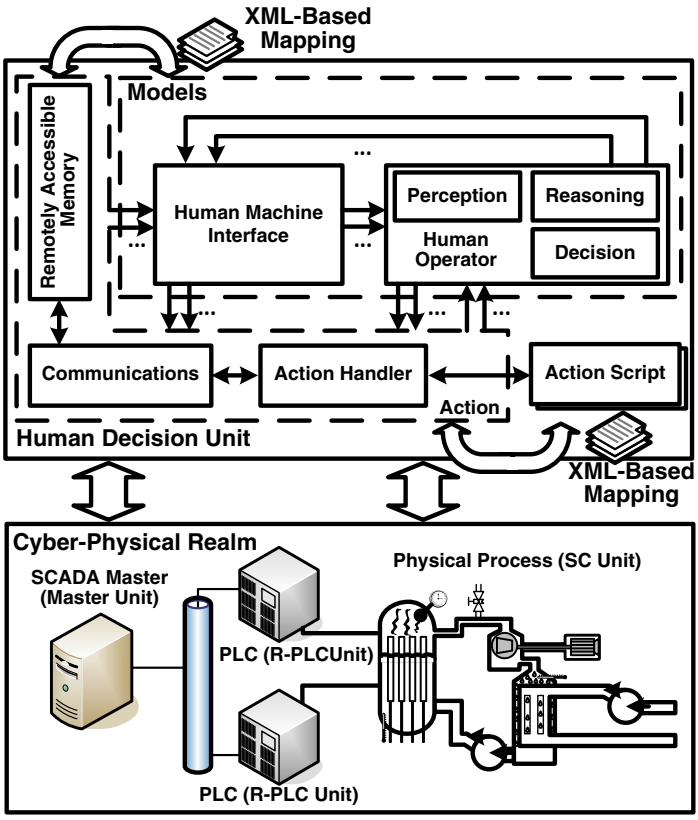


Figure 2. Extended architecture with a human decision unit.

structed according to the specifications of an experiment, three building blocks are included in the human operator model for completeness: (i) perception; (ii) reasoning; and (iii) decision. These basic human operator functionalities can be taken into account when building similar models. Also, depending on the requirements of an experiment, they can be replaced with other blocks if necessary.

As shown in Figure 2, human operator models receive events from the action module directly and through HMI models. The first case represents the direct interaction of human operators with the cyber-physical realm, while the second case represents interactions through the HMI. In both cases, the inputs are the data used by the models in each time step (e.g., measured voltage) while the outputs are the actions to be executed. Each action includes an identifier and several parameters (e.g., open or close a valve). These are written to the remotely accessible memory module from where they are read by the action

handler module. Then, based on external XML configuration files, the action handler module runs a specific action script identified by a numeric identifier.

Action scripts are written in the C programming language and are loaded as external binary libraries. These can be specific to each experiment and may include commands that pass values to other components of the framework (e.g., physical model) or commands that launch additional scripts (e.g., configuring a firewall or turning off a machine). This way, the HD unit provides a flexible approach to translate model-specific outputs to real actions that affect the physical and cyber realms.

The remotely accessible memory module is not limited to passing values between the internal modules of the HD unit; it also serves as a way for external components to interact with the human operator and HMI models. By enabling remote access to this memory region, external software components (e.g., other HD units and SCADA master units) can communicate with the HD unit using simple memory access operations. In these cases, requests are received and processed by the communications module, which passes the received values to the remotely accessible memory module, from where they are read by the models module and provided to the human operator and HMI models. To increase flexibility, external XML configuration files are used to map memory regions to model inputs and to map model outputs to memory regions.

As in our previous work, we implemented a prototype of the HD unit in C# (Windows) and ported it to Unix-based systems using the Mono platform. Currently, communications with other HD and master units are handled by the Modbus protocol, as this protocol was specifically designed for exchanging memory-mapped data between units. However, other protocols can be implemented simply by replacing the Modbus handler units.

Human operator and HMI models were implemented in Matlab Simulink, a general simulation environment for dynamic and embedded systems. Its toolboxes (e.g., control systems and neural networks) provide powerful support for modeling and simulation. Matlab RTW was used to generate C code from the Simulink models. The generated code was then integrated into the extended framework using an XML configuration file. Thus, the model is able to interact in real time with the other system components.

5. Case Study

Human operators play major roles in real critical infrastructure installations and, therefore, cannot be ignored in experimentation testbeds. The HD units address this issue by bringing new elements that implement the human-in-the-loop paradigm in existing testbeds. This section presents experimental results that demonstrate the applicability of the proposed approach and the importance of human operators in a case study involving cyber attacks on a simulated power grid.

The power grid model employed in the experiment was the well-known IEEE 30-bus test system, which includes six power generators and twenty loads distributed on twenty buses. The overall model was divided into four regions,

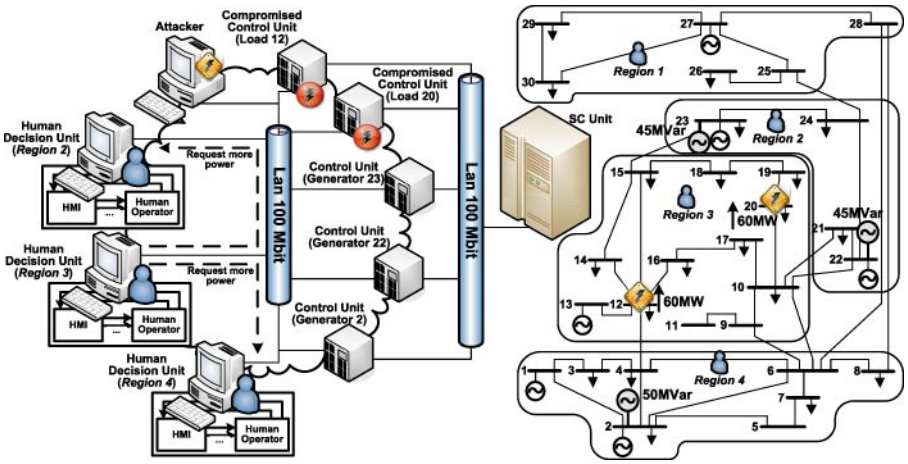


Figure 3. Experimental setup and test system regions.

each controlled by different human operators. One of the main goals of the experiment was to show that disturbances can be balanced by countermeasures taken by power grid operators. The results also confirm the fact that, in an interconnected power grid, operators from different regions must collaborate to restore the grid to normal conditions after a disturbance.

5.1 Experimental Scenarios

Figure 3 presents a schematic diagram of the IEEE 30-bus power grid model. The grid is divided into four regions that minimize the number of connections (i.e., transmission lines) between the regions. In a real-world environment, the grid may be divided into regions based on other factors (e.g., balanced power and loads). However, our focus is not on defining regions, but on recreating a possibly realistic scenario with multiple operators, who have to cooperate in order to keep the power grid within its normal operating limits.

In our scenario, we assume that an adversary can compromise the information and communications technology infrastructure in Region 3 using social engineering and also exploit vulnerabilities in the SCADA protocols [2]. Then, the same adversary employs a coordinated worm-based attack to trigger synchronized events in control devices. The attacks turn the power on to large consumers at the same time at Substations 12 and 20 (120 MW total), which causes a disturbance that drops the voltages below their operating limits of 0.95 p.u. In a real-world environment, these attacks could damage hardware and cause blackouts, and possibly have cascading effects that could spread throughout the power grid.

In our scenario, we also assume that the operators in Region 3 can request assistance from operators in other regions when they lose control of their in-

frastructure. Upon receiving the request, operators in Regions 2 and 4 inject additional power into the grid with the goal of stabilizing the voltages in Region 3.

The following three cases are considered in our scenario:

- **Case 1:** Operators in Region 3 lose control of their infrastructure and do not request assistance from operators in other regions. The voltages drop well below their normal operating limits and the operators have to react quickly in order to prevent serious damage to physical devices.

- **Case 2:** Operators in Region 3 lose control of their infrastructure and request assistance from operators in Region 4. This case shows that a disturbance originating in Region 3 can be addressed by measures taken in other regions. The operators in Region 3 request their counterparts in Region 4 to increase energy production and the power supplied to the grid. Upon receiving the request, the operators in Region 4 start their back-up generators and increase production by 50 MVar. Although the effects are limited, this case demonstrates the importance of operator collaboration.

- **Case 3:** Operators in Region 3 lose control of their infrastructure and request assistance from operators in Regions 4 and 2. The actions taken in Case 2 produce limited effects, so additional assistance is requested from operators in Region 2. Upon receiving the request, the operators in Region 2 increase the energy production by 90 MVar. This action stabilizes the bus voltages in Region 3.

5.2 Experimental Setup and Models

The critical infrastructure protection scenario was implemented in the Experimental Platform for Internet Contingencies (EPIC) Laboratory at the EU Joint Research Centre [20]. The Emulab testbed included nodes with the following configuration: FreeBSD OS 8, AMD 2.3 GHz Athlon Dual Core CPU and 4 GB RAM. Figure 3 shows the experimental setup.

Models were developed using the "black-box" approach described in the previous sections. A Simulink model was developed for each operator with inputs and outputs connected to action scripts.

The operator model for Region 3 has one input (link status) and three outputs (action identifier and two parameters for the request for assistance). The input action script continuously tests the link between the HD unit and the physical process (SC unit). When the link is "on," the action script sends a value of "1" to the model, and "0" otherwise. The model takes this input and forwards its negated value to the output along with the action identifier. The output action script takes these two values and forwards them to the HD units running in Regions 2 and 4. In the implementation, a value of "1" means that operators in Region 3 need assistance. The implemented functions are different for each case:

$$\begin{aligned}
&\text{Case 1:} \quad f_1(x) = (ID, 0, 0)\\
&\text{Case 2:} \quad f_2(x) = (ID, !x, 0)\\
&\text{Case 3:} \quad f_3(x) = (ID, !x, !x)
\end{aligned}$$

where x is the model input and ID is the action identifier.

The operator model for Region 4 has one input (status of assistance request ("0" or "1")) and two outputs (action identifier and one parameter denoting the power injected by back-up generators (50 MVar)). The model receives its input value from the HD unit in Region 3 and produces an output that is sent to the SC unit and finally to the physical process. The mathematical function for this model is:

$$f(x) = (ID, (x = 1)?\ 50 : 0)$$

where x is the model input and ID is the action identifier.

The operator model for Region 2 is similar to that for Region 4 in that it takes the same input but produces one additional output. The first output is the action identifier, while the second and the third outputs are the MVars produced by two back-up generators. Our scenario uses two back-up generators in Region 2 to produce a total of 90 MVar (45 MVar per generator). The mathematical function for this model is:

$$f(x) = (ID, (x = 1)?\ 45 : 0, (x = 1)?\ 45 : 0)$$

where x is the model input and ID is the action identifier.

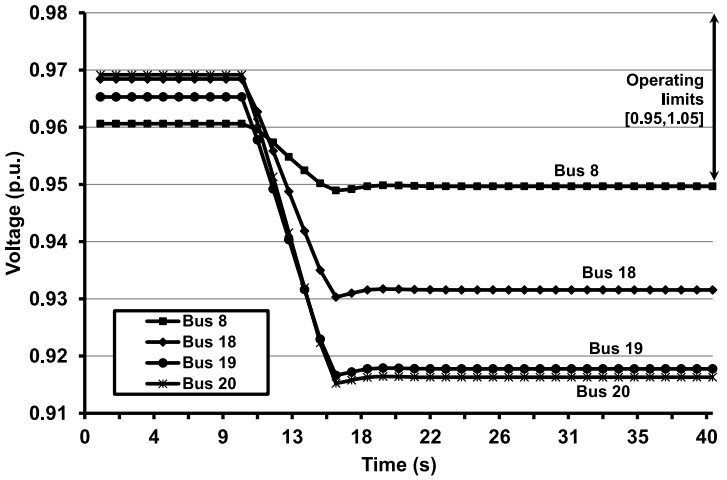
5.3 Experimental Results

This section presents the results obtained for the three cases.

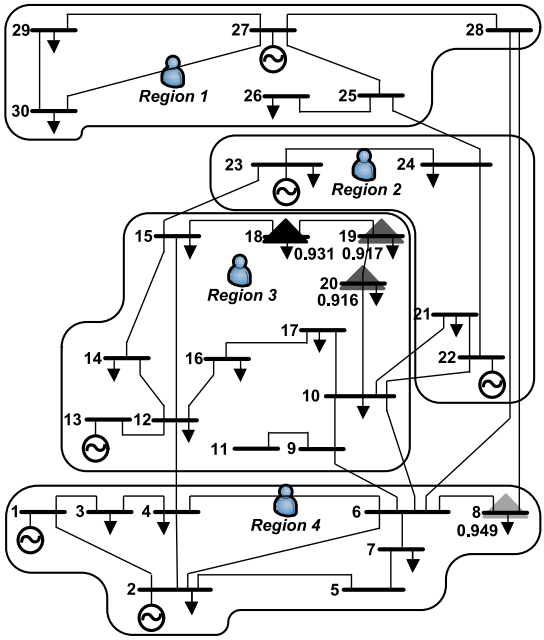
- **Case 1:** Immediately after the attack is launched on Buses 12 and 20, the voltages begin to drop. The voltages of the Buses 18, 19 and 20 fall to almost 0.91 p.u., well below the operating limit of 0.95 p.u. The disturbance also propagates to Region 4, where it causes a voltage drop on Bus 8 to 0.949 p.u., slightly below the operating limit. This effect is also shown in Figures 4(a) and (b) where, without any operator intervention, the disturbance causes severe voltage changes, mostly in Region 3.

- **Case 2:** In this case, we assume that the operators in Region 3 are able to obtain assistance from their counterparts in Region 4, where an additional back-up generator injects 50 MVar into the grid. This action has a significant effect on Bus 8, where the voltage increases above the operating limit. However, the effects are not as significant on the other buses, where the voltages remain below 0.95 p.u. (Figure 5).

- **Case 3:** In Case 3, the operators in Region 3 request the assistance of operators in Regions 4 and 2. The operators in Region 4 inject 50 MVar



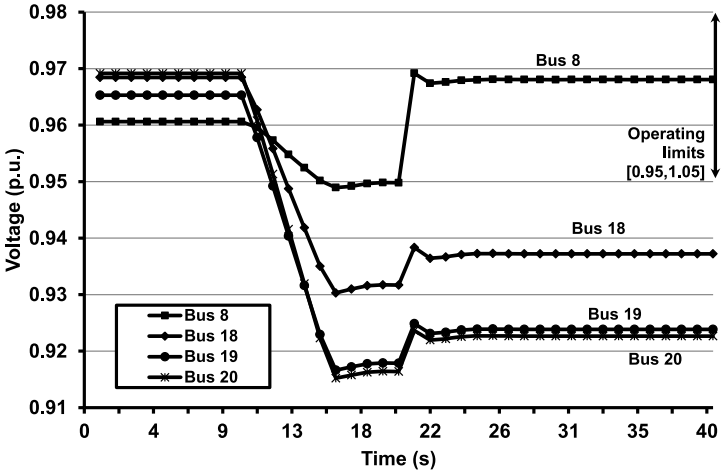
(a) Time series of affected bus voltages.



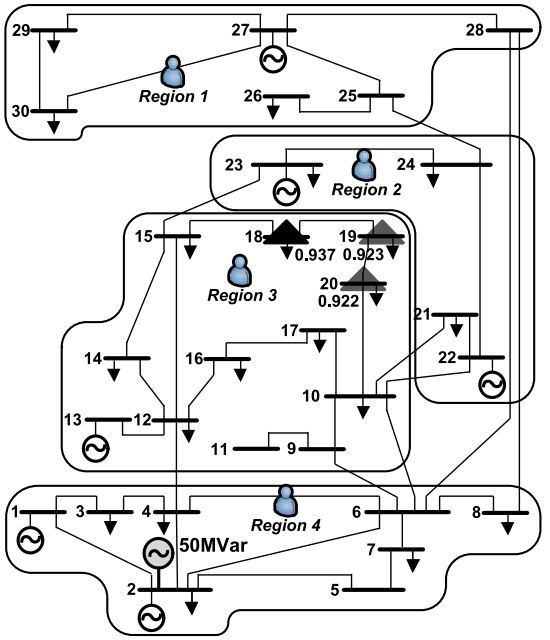
(b) Regional view.

Figure 4. Effect of the attack in Case 1.

of additional power using a back-up generator and the operators in Region 2 start two back-up generators, which inject an additional 90 MVar of power into the grid. As shown in Figure 6, this immediately increases



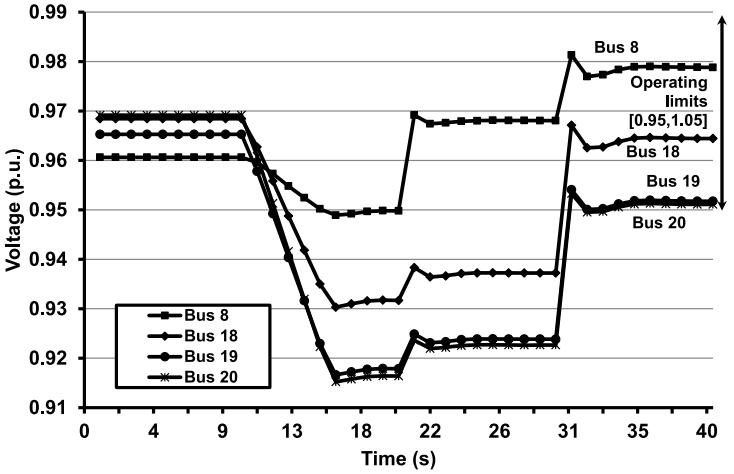
(a) Time series of affected bus voltages.



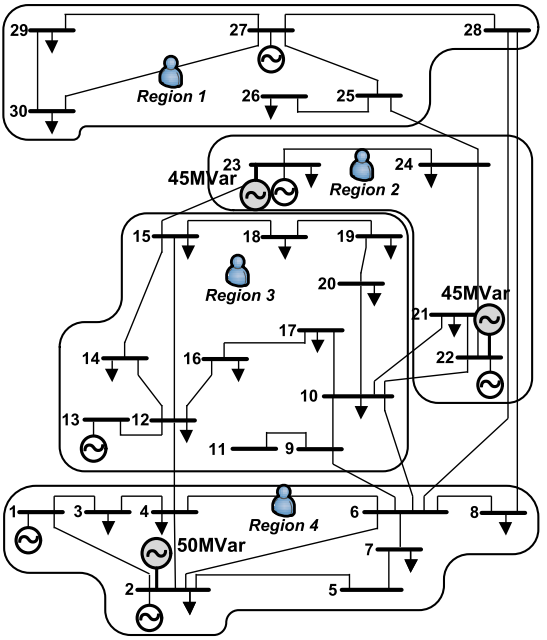
(b) Regional view.

Figure 5. Effect of the attack in Case 2.

the voltages above the operating limits. Consequently, although the operators in Region 3 lose control of their infrastructure, they are eventually



(a) Time series of affected bus voltages.



(b) Regional view.

Figure 6. Effect of the attack in Case 3.

able to ensure stability in their region by cooperating with operators in neighboring regions.

The results confirm the fact that human operators are indispensable elements of security studies involving critical infrastructures. Upon comparing the results for Case 3 with those for Cases 1 and 2, it is clear that operators in a highly interconnected power grid must collaborate in order to balance regional disturbances and ensure grid stability. The results also demonstrate that the proposed approach can recreate complex scenarios involving the cyber and physical realms, as well as the decision making of human operators. The approach thus implements the important human-in-the-loop paradigm that is a significant aspect of real-world critical infrastructure environments.

6. Conclusions

Human operators play important roles in supervising and controlling modern critical infrastructures. Although industry may be moving towards fully automated control loops, human operators are indispensable during abnormal situations and contingencies. The principal contribution of this paper is an extended experimentation framework that provides generic human decision units that help integrate human operator and HMI models. The integration is based on a "black-box" approach where, as long as generic models implement well-defined interfaces with sets of input and output signals, the models can be integrated in the experimentation framework regardless of their content. The case study involving the IEEE 30-bus power grid model demonstrates the utility of the extended experimentation framework. The results of the case study also confirm that, in large-scale interconnected critical infrastructures involving multiple operators, it is crucial that operators cooperate to ensure the global stability of the infrastructures. Therefore, modern testbeds must provide support for modeling and analyzing the behavior of multiple human operators in complex critical infrastructure scenarios.

Our future research will explore the complexity of human operator networks and the vulnerabilities of critical infrastructures that rely on information exchange. This will help recreate and analyze complex multi-sector scenarios where one critical infrastructure could have cascading effects on other critical infrastructures.

References

[1] G. Bekey and C. Neal, Identification of sampling intervals in sampled-data models of human operators, *IEEE Transactions on Man-Machine Systems*, vol. 9(4), pp. 138–142, 1968.

[2] E. Bompard, P. Cuccia, M. Masera and I. Nai Fovino, Cyber vulnerability in power systems operation and control, in *Critical Infrastructure Protection (LNCS 7130)*, J. Lopez, R. Setola and S. Wolthusen (Eds.), Springer-Verlag, Berlin Heidelberg, Germany, pp. 197–234, 2012.

[3] R. Chabukswar, B. Sinopoli, G. Karsai, A. Giani, H. Neema and A. Davis, Simulation of network attacks on SCADA systems, presented at the *First Workshop on Secure Control Systems*, 2010.

[4] K. Craik, Theory of the human operator in control systems, *British Journal of Psychology*, vol. 38(2), pp. 56–61, 1947.

[5] C. Davis, J. Tate, H. Okhravi, C. Grier, T. Overbye and D. Nicol, SCADA cyber security testbed development, *Proceedings of the Thirty-Eighth North American Power Symposium*, pp. 483–488, 2006.

[6] B. Genge, I. Nai Fovino, C. Siaterlis and M. Masera, Analyzing cyber-physical attacks on networked industrial control systems, in *Critical Infrastructure Protection V*, J. Butts and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 167–183, 2011.

[7] A. Gheorghe, M. Masera, M. Weijnen and L. De Vries, *Critical Infrastructures at Risk: Securing the European Electric Power System*, Springer, Dordrecht, The Netherlands, 2006.

[8] R. Harris, J. Kaplan, C. Bare, H. Iavecchia, L. Ross, D. Scolaro and D. Wright, Human Operator Simulator (HOS) IV User's Guide, Research Product 89-19, U.S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia, 1989.

[9] K. Hopkinson, K. Birman, R. Giovanini, D. Coury, X. Wang and J. Thorp, EPOCHS: Integrated commercial off-the-shelf software for agent-based electric power and communication simulation, *Proceedings of the 2003 Winter Simulation Conference*, vol. 2, pp. 1158–1166, 2003.

[10] T. Ivancevic and B. Jovanovic, Human operator modeling and Lie-derivative based control (`arxiv.org/pdf/0907.1206.pdf`), 2009.

[11] M. McDonald, G. Conrad, T. Service and R. Cassidy, Cyber Effects Analysis Using VCSE: Promoting Control System Reliability, Technical Report SAND2008-5954, Sandia National Laboratories, Albuquerque, New Mexico and Livermore, California, 2008.

[12] F. Moussa, C. Kolski and M Riahi, A model based approach to semi-automated user interface generation for process control interactive applications, *Interacting with Computers*, vol. 12(3), pp. 245–279, 2000.

[13] I. Nai Fovino, M. Masera, L. Guidi and G. Carpi, An experimental platform for assessing SCADA vulnerabilities and countermeasures in power plants, *Proceedings of the Third Conference on Human System Interaction*, pp. 679–686, 2010.

[14] S. Neema, T. Bapty, X. Koutsoukos, H. Neema, J. Sztipanovits and G. Karsai, Model-based integration and experimentation of information fusion and C2 systems, *Proceedings of the Twelfth International Conference on Information Fusion*, pp. 1958–1965, 2009.

[15] PowerWorld Corporation, Champaign, Illinois (`www.powerworld.com`).

[16] C. Queiroz, A. Mahmood, J. Hu, Z. Tari and X. Yu, Building a SCADA security testbed, *Proceedings of the Third International Conference on Network and System Security*, pp. 357–364, 2009.

[17] A. Rao and M. Georgeff, BDI agents: From theory to practice, *Proceedings of the First International Conference on Multi-Agent Systems*, pp. 312–319, 1995.

[18] R. Reeder and R. Maxion, User interface defect detection by hesitation analysis, *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 61–72, 2006.

[19] X. Shi and Q. Zhong, The introduction of high level architecture (HLA) and run-time infrastructure (RTI), *Proceedings of the Society of Instrument and Control Engineers Annual Conference*, vol. 1, pp. 1136–1139, 2003.

[20] C. Siaterlis, A. Garcia and B. Genge, On the use of Emulab testbeds for scientifically rigorous experiments, to appear in *IEEE Communications Surveys and Tutorials*.

[21] C. Wang, L. Fang and Y. Dai, A simulation environment for SCADA security analysis and assessment, *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation*, pp. 342–347, 2010.

[22] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb and A. Joglekar, An integrated experimental environment for distributed systems and networks, *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation*, pp. 255–270, 2002.

[23] X. Zhao, J. Venkateswaran and Y. Son, Modeling human operator decision-making in manufacturing systems using BDI agent paradigm, presented at the *IIE Annual Conference and Exposition*, 2005.