PolyPublie
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Defending the SCADA Network Controlling the Electrical Grid from Advanced Persistent Threats
Auteur: Author:	Antoine Lemay
Date:	2013
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Lemay, A. (2013). Defending the SCADA Network Controlling the Electrical Grid from Advanced Persistent Threats [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1300/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/1300/
Directeurs de recherche: Advisors:	Scott Knight, & Jose Manuel Fernandez
Programme: Program:	Génie informatique

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

DEFENDING THE SCADA NETWORK CONTROLLING THE ELECTRICAL GRID FROM ADVANCED PERSISTENT THREATS

ANTOINE LEMAY

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION

DU DIPLÔME DE PHILOSOPHIÆ DOCTOR

(GÉNIE INFORMATIQUE)

DÉCEMBRE 2013

© Antoine Lemay, 2013.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

DEFENDING THE SCADA NETWORK CONTROLLING THE ELECTRICAL GRID FROM ADVANCED PERSISTENT THREATS

présentée par : LEMAY Antoine

en vue de l'obtention du diplôme de : Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. QUINTERO Alejandro, Doct., président

M. FERNANDEZ José M., Ph.D., membre et directeur de recherche

M. KNIGHT Scott., Ph.D., membre et codirecteur de recherche

M. KOCAR Ilhan, Ph.D., membre

M. MORRIS Thomas H., Ph.D., membre

ACKNOWLEDGEMENTS

The completion of this research would not have been possible without the help of a large number of people. I would like to start by thanking the research interns that directly contributed to this project: Bernard Rosset and Étienne Ducharme. Their invaluable work greatly facilitated the implementation of a number of concepts and, as such, were integral to the success of the project. The work of Pier-Luc St-Onge, former analyst at the SecSI lab, may not be glamorous, but must also be recognized as an integral part of this project.

I would also like to thank Prof. Fernandez and Prof. Knight who provided me with valuable guidance and incredible opportunities during the course of this project and managed to tolerate my incessant argumentation for the duration of this project. My gratitude also extends to Dr. de Jesus who was very generous with his insights and his time. I hope he realizes his dream of protecting the critical infrastructure.

I also take the opportunity to salute my colleagues at the SecSI lab in Montreal and at the CSL lab in Kingston who provided both valuable feedback and constant comradeship. In particular, I would like to underline the contribution of Joan Calvet who proved it was indeed possible to graduate from this place.

I would be remiss if I did not acknowledge the contribution of the agencies that provided the financing for this project. In particular, the ISSNet network provided funding for both the research and the frequent travel between Montreal and Kingston. Without their support, the amazing ability to get the best of both worlds would not have been possible. The NSERC also contributed to the success of this research project by providing the funding for the acquisition of specialized SCADA equipment without which none of the engineering work would have been possible.

Finally, I wish to extend my deepest gratitude to my family: my father Guy, my mother Nicole and my brother Guillaume who provided their support, even if it usually took the form of snide remarks. However, all of this pales in comparison to the debt I owe to Marie-Hélène who knows the burden of pursuing a Ph.D. and made sure I had the all the love required to follow this dream. So, a big "Thank you".

RÉSUMÉ

Les civilisations modernes sont dépendantes des technologies de l'information et des communications. Par ce fait, elles requièrent une alimentation constante en électricité pour assurer leur prospérité. Un siècle de travaux acharnés par des ingénieurs en électronique de puissance permet de garantir la fiabilité des réseaux électriques. Un des outils pour arriver à cette fin est une augmentation de l'automatisation et du contrôle à distance des réseaux électriques. Cette technologie permet aux contrôleurs qui opèrent le réseau électrique d'ajuster automatiquement des paramètres opérationnels pour faire face aux contraintes extérieures au fur et à mesure que ces contraintes évoluent. Par exemple, une augmentation de la demande suite à une vague de froid va automatiquement entraîner une augmentation de l'approvisionnement par l'envoi de commandes à distance pour ouvrir les vannes à la centrale hydroélectrique et faire tourner les turbines plus rapidement. Ceci garanti que le réseau électrique fonctionne toujours à pleine capacité et livre l'énergie électrique avec fiabilité, sans égard aux conditions externes.

Paradoxalement, les gains offerts par les systèmes automatisés ont introduit un risque jusqu'alors inconnu à la fiabilité du réseau électrique : les cyber attaques. Pour permettre l'automatisation, les opérateurs de réseaux électriques se sont tournés vers la technologie d'acquisition de données et de supervision, mieux connu sous le nom de système SCADA. De nos jours, la technologie SCADA se base sur du matériel et des logiciels commerciaux comme les communications TCP/IP via Ethernet ou comme le système d'exploitation Windows. Ceci permet aux entités malicieuses de faire usage de leur savoir concernant les techniques offensives qu'ils ont développé pour attaquer les systèmes traditionnels faisant usage de ces technologies.

La majorité de ces entités sont des menaces diffuses cherchant principalement à acquérir de la capacité de stockage servant à héberger du contenu illégal, du temps machine pour envoyer du spam ou des mots de passe pour permettre la fraude. Cet objectif est plus facile à atteindre en attaquant des ordinateurs personnels plutôt que des machines d'un réseau SCADA. Toutefois, certains acteurs ciblent délibérément les réseaux SCADA puisque ceux-ci ont le potentiel de causer des dégâts dans le monde physique. Ces acteurs recherchent agressivement les vulnérabilités et persévèrent dans leurs attaques, même face à une amélioration de la capacité défensive du réseau. Ces acteurs se font affubler le qualificatif de menaces persistantes avancées

ou APTs. À cause de cette volonté de cibler un réseau spécifique, il est plus difficile de détourner ces attaquants vers d'autres victimes.

Si nous souhaitons empêcher ces APTs de s'attaquer aux réseaux SCADA qui contrôlent l'infrastructure critique, nous devons élaborer une stratégie qui ne repose pas sur la réduction complète des vulnérabilités. Un bon nombre de contraintes opérationnelles, comme le mode d'opération 24/7 qui rend la tenue de périodes de maintenance difficile, garantissent qu'il y aura toujours au moins une vulnérabilité potentiellement exploitable par un attaquant.

Dans ce contexte, l'objectif de ce projet de recherche est d'aider les opérateurs de réseaux électriques à défendre leur réseau SCADA contre les menaces persistantes avancées. Pour atteindre cet objectif, nous visons à mieux comprendre comment le comportement des menaces persistantes avancées se manifeste dans un réseau SCADA et à développer, en se basant sur des preuves expérimentales, de nouveaux outils et techniques pour se défendre contre les comportements attendus.

En analysant les travaux antérieurs, on reconnaît que la vraie nature d'un réseau SCADA est de servir de boucle de contrôle pour le réseau électrique. Une conséquence directe est que tout attaquant qui obtient accès au réseau SCADA peut altérer l'état du réseau électrique à sa guise. Si un APT voudrait poursuivre ce but, la recherche actuelle en sécurité des réseau SCADA ne parviendrait pas à prévenir cette attaque puisqu'elle n'est pas orientée vers stopper les attaquants hautement qualifiés. Ceci rend les réseaux SCADA invitants pour les états engagés dans une compétition agressive. Malgré cela, aucun cyber incident majeur causant des dégâts physiques n'est répertorié à ce jour.

En se basant sur cette observation, nous avons développé un modèle d'attaque pour le comportement d'un APT dans un réseau SCADA qui n'implique pas nécessairement des dommages massifs dans le monde physique. Ainsi, nous avons introduit le scénario d'attaque par trou d'aiguilles, notre première contribution majeure, dans lequel un attaquant cause de petits dégâts qui s'accumulent sur une longue période pour éviter d'être détecté.

À partir de ce scénario, nous avons développé une stratégie consistant à augmenter la capacité de surveillance, c'est-à-dire de renforcer la puissance de la détection, pour prévenir l'utilisation de ce scénario d'attaque par les APTs. En se basant sur notre intuition que la détection d'intrusion par

anomalie sera particulièrement efficace dans le contexte hautement régulier d'un réseau SCADA, l'utilisation de cette technique est favorisée.

Pour tester les capacités de notre détecteur, nous devons adresser le problème du manque d'infrastructures expérimentales adaptées à la recherche en sécurité des réseaux SCADA. Une revue de la littérature montre que les approches expérimentales courantes ne sont pas appropriées pour générer des données réseau avec une haute fidélité. Pour résoudre ce problème, nous avons introduit le concept du *Carré de sable ICS*, notre deuxième contribution majeure, qui utilise une approche hybride combinant la haute fidélité des résultats de l'émulation et le facteur d'échelle et le faible coût de la simulation pour créer un montage expérimental capable de produire des données réseau de haute fidélité, adaptées à l'usage expérimental.

Finalement, nous avons été en mesure de tester une implémentation d'un système de détection d'intrusion par anomalies, notre troisième contribution majeure, en utilisant le *Carré de sable ICS*. En utilisant des caractéristiques simples, il est possible de détecter du trafic de commandement et contrôle dans un réseau SCADA, ce qui force les attaquant à utiliser pour leurs opérations routinières de maintenance de complexes canaux cachés dont la bande passante est limitée. Ceci atteste de la validité de notre intuition selon laquelle la détection par anomalie est particulièrement efficace dans les réseaux SCADA, revitalisant par le fait même une technique de défense qui a longtemps été délaissée à cause de sa piètre performance dans les réseaux corporatifs typiques.

La somme de ces contributions représente une amélioration significative de l'état de la défense des réseaux SCADA contre les menaces persistantes avancées, incluant les menaces en provenance des services de renseignement étatiques. Ceci contribue à une augmentation de la fiabilité des infrastructure critiques, et des réseaux électriques en particulier, face à un intérêt grandissant de la part des cyber attaquants.

ABSTRACT

Modern civilization, with its dependency on information technology, require a steady supply of electrical power to prosper. A century of relentless work by power engineers has ensured that the power grid is reliable. One of tools they used to achieve that goal is increased automation and remote control of the electrical grid. This technology allows the controllers supervising the power grid to automatically adjust operational parameters to meet external constraints as they evolve. A new surge in demand from a cold night will trigger an automated increase in supply. Remote control commands will be sent to open sluice gates at the hydroelectric plant to make turbines spin faster and generate more power. This ensures the electric grid always functions at peak efficiency and reliably deliver power no matter what the external conditions are.

Paradoxically, the gains provided by the automated systems invited a previously unknown risk to the reliability of power delivery: cyber attacks. In order to achieve automation, utility operators have turned to Supervisory Control and Data Acquisition, or SCADA, technology. In this era, SCADA technology is built on top of commercial off the shelf hardware and software such as TCP/IP over Ethernet networks and Windows operating system. This enables malicious entities to leverage their pre-existing knowledge of offensive techniques known to work on these platform to attack the SCADA networks controlling critical infrastructure.

Of those entities, the majority are unfocused attackers searching for commodity assets such as storage capacity to store illegal materials, processing power to send spam or credentials to enable fraud. However, some actors are deliberatively targeting the SCADA networks for their ability to cause damage in the physical realm. These actors aggressively search for vulnerabilities and are stubborn in the face of an increase in defensive measures and are dubbed advanced persistent threats, or APTs. As such, it is more difficult to turn them away.

If we want to prevent these advanced persistent threats from preying on the SCADA networks controlling our critical infrastructure, we need to devise a defense that does not rely on completely removing vulnerabilities. A number of operational constraints, such as the need to operate 24/7 precluding the opening of maintenance windows, ensure that there will always be a vulnerability that can be exploited by an attacker.

In that light, the goal of this research project is to is to help power grid operators defend their SCADA networks against advanced persistent threats. To achieve that goal we aim to better understand how the behaviour of advanced persistent threats will manifest itself in a SCADA network and to develop, based on evidence derived from experiments, new tools and techniques to defeat the expected behaviour.

By analyzing prior work, we recognize that the true nature of SCADA networks is to serve as a basic control loop for the electric grid. A direct consequence is that any attacker gaining access to the SCADA network could send the grid into any state he wishes. We also showed that, should advanced persistent threats attempt to pursue this goal, current research in SCADA security would not provide significant help, not being focused on preventing the exploitation of SCADA network by skilled attackers. This makes SCADA networks attractive to nation states engaged in aggressively competitive behaviour. However, no evidence of major cyber incidents causing physical damage is forthcoming.

From that observation, we developed an attacker model for advanced persistent threat behaviour in SCADA networks that did not necessarily involve causing massive physical damage. So, we introduced the pinprick attack scenario, our first major contribution, in which an attacker causes small amounts of damage that accumulate over time in order to stay under the radar.

From this scenario, we developed a strategy of increasing the capability of surveillance, or boosting the radar so to speak, in order to prevent advanced persistent threats from using this scenario. The use of anomaly-based intrusion detection was favored based on our intuition that it would prove very effective in the highly regimented context of SCADA networks.

To test the capability of our detector, we needed to address the lack of experimental infrastructure suitable for network security. However, a study of the literature shows that current experimental approaches are not appropriate to generate high fidelity network data. To solve this problem, we introduced the ICS sandbox concept, our second major contribution, that used a hybrid approach combining the high fidelity results of emulation and the scalability and cost reduction of simulation to create an experimental setup able to produce high fidelity network data sets for experimentation.

Finally, we were able to test an implementation of anomaly-based intrusion detection, our third major contribution, using the ICS sandbox. Using only simple features, it was possible to detect

command and control traffic in a SCADA network and push attackers to use complex covert channels with limited bandwidth to perform their routine maintenance operations. This attests to the validity of our intuition that anomaly-based detection is particularly effective in SCADA network, revivifying a defensive technique that suffers from poor performance in typical corporate networks.

The sum of these contributions represent a significant improvement in the defense of SCADA networks against advanced persistent threats, including threats from nation state sponsored intelligence agencies. This contributes to the increased reliability of critical infrastructure, and of the electrical grid in particular, in the face of an increasing interest by cyber attackers.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS III

RÉSUMÉ IV

ABSTRACT VII

TABLE OF CONTENTS X

LIST OF TABLES XV

LIST OF FIGURES XVI

LIST OF ACRONYMS AND ABBREVIATIONS XVIII

CHAPTER 1 INTRODUCTION 1

1.1 Current state of SCADA network security 3

1.1.1 Vulnerability of SCADA systems 3

1.1.2 Policy efforts to address the vulnerability 7

1.1.3 SCADA related incidents 9

1.2 Problem definition 14

1.2.1 Research goal 14

1.2.2 Research aim 16

1.2.3 Research objectives 17

1.3 Thesis organization 19

CHAPTER 2 LITTERATURE REVIEW 21

2.1 The power grid 22

2.1.1 Power line 22

2.1.2 Substation 23

2.1.3 Overview of the power grid 24

2.2 Elements of control 26

2.2.1 Basic control theory 26
2.2.2 Application to the electric grid 27
2.2.3 Control center 29
2.3 SCADA system architecture 32
2.3.1 SCADA system architecture 32
2.3.2 Equipment examples 34
2.3.3 DNP3 protocol 37
2.4 Experimental approaches in SCADA experimentation 41
2.4.1 Full Physical Deployment 41
2.4.2 Partial implementation 42
2.4.3 Software only 43
2.4.4 Simulation 43
2.4.5 Emulation 45
2.4.6 Impact assessment 45
2.5 SCADA security research 48
2.5.1 Offensive research 48
2.5.2 Preventing attacks in SCADA systems 50
2.5.3 Detecting attacks in SCADA systems 52
2.6 Conclusion 54
CHAPTER 3 BEHAVIOUR OF ADVANCED PERSISTENT THREATS 57
3.1 Attacking the critical infrastructure 58
3.1.1 Choosing critical infrastructure as a target 58
3.1.2 Low probability/high impact scenario 60
3.1.3 From cyber warfare to cyber conflict 60

3.2 Cyber conflict model 61
3.2.1 Pinprick attacks 62
3.2.2 The case of Stuxnet 64
3.2.3 Raising the bar 65
3.3 Defensive strategy 66
3.3.1 The use of covert communication by advanced attackers 67
3.3.2 Communication model 67
3.4 conclusion 71
CHAPTER 4 IMPROVING THE FIDELITY OF SCADA NETWORK SECURITY EXPERIMENTAL METHODS 72
4.1 The ICS sandbox 72
4.1.1 Scoping 73
4.1.2 Implementation 74
4.2 Validation with the ICS sandbox SCADA emulation component 79
4.2.1 Description of the training 79
4.2.2 Evaluation and lessons learned 81
4.3 Validation of the ICS sandbox simulation/emulation approach 82
4.3.1 IEEE reliability test system 83
4.3.2 Terrorist threat experiment 87
4.3.3 Adapting the ICS sandbox 88
4.3.4 Reproducing the scenario 93
4.4 Conclusion 95
CHAPTER 5 ANOMALY-BASED INTRUSION DETECTION IN SCADA NETWORKS 97
5.1 Methodology 98
5.1.1 Characterizing SCADA traffic 98

5.1.2 Experimental Setup 99
5.2 A portrait of "normal" 101
5.2.1 Feature selection 102
5.2.2 Logical Topology 104
5.2.3 Interdeparture time 105
5.2.4 Packet size 107
5.2.5 Sensitivity analysis 110
5.3 Scenarios 113
5.3.1 Common botnet 114
5.3.2 Advanced persistent threat 115
5.3.3 Covert channel 116
5.4 Results 118
5.4.1 Scenario 1 - Botnet 118
5.4.2 Scenario 2 -APT 122
5.4.3 Scenario 3 - covert channel 126
5.4.4 Entropy measurements for covert channel 129
5.5 Conclusion 133
CHAPTER 6 CONCLUSION 135
6.1 Behaviour of advanced persistent threats 136
6.2 The ICS Sandbox 138
6.2.1 Contribution 138
6.2.2 Limitations 139
6.2.3 Future work 141
6.3 Anomaly-based detection in SCADA networks 143

6.3.1 Contribution 143

6.3.2 Limitations 145

6.3.3 Future work 146

CHAPTER 7 BIBLIOGRAPHY 148

LIST OF TABLES

Table 4-1: Power flow calculation example....85

Table 5-1: Wireshark conversation analysis 104

Table 5-2: Sensitivity analysis - number of points....112

Table 5-3: Sensitivity analysis - number of RTUs....112

Table 5-4: Conversation analysis with infected RLS (Hardcoded Waledac C&C in red)....119

Table 5-5: Conversation analysis with pivot point (malicious conversations in red) 122

Table 5-6: Conversation list - covert channel experiment 9 LSBs 126

LIST OF FIGURES

Figure 1-1 : Skill and motivation of threat agents in industrial control 8

Figure 1-2: Stuxnet communications 11

Figure 1-3 : Number of disclosed SCADA product vulnerabilities 12

Figure 2-1 : Power flow through a typical substation 23

Figure 2-2 : Overview of the grid 25

Figure 2-3 : Basic control loop 26

Figure 2-4 : Control loops for the electric grid 28

Figure 2-5 : SCADA control loop 29

Figure 2-6 : Control Center functions 31

Figure 2-7 : SCADA architecture for the power grid 34

Figure 2-8: Example MTU HMI 35

Figure 2-9: RTU example 36

Figure 2-10: Example PLC 37

Figure 2-11: DNP3 layers 39

Figure 2-12: DNP3 over IP 40

Figure 3-1: Spectrum of warfare 61

Figure 3-2: Communication model 68

Figure 3-3: Shannon-based communication model 69

Figure 4-1: ICS Sandbox architecture 74

Figure 4-2: Black box design of simulator and SCADA modules 78

Figure 4-3: Training network infrastructure 80

Figure 4-4: IEEE reliability test system network 83

Figure 4-5: Simplified distribution bus 84

Figure 4-6: Near-best interdiction plans proposed by Salmeron et al. 88

Figure 4-7: Black box architecture using PyPower as power simulation 91

Figure 4-8: Impact of the interdictions from scenario 1 on generation cost 94

Figure 5-1 : Experiment network 100

Figure 5-2 : Localized simulator design 101

Figure 5-3: Average Interdeparture time for the MTU by RTU 105

Figure 5-4: Interdeparture time MTU - multiplexed 106

Figure 5-5: Interdeparture time RTU 107

Figure 5-6: Distribution of packet lengths - MTU 108

Figure 5-7: Distribution of packet lengths - RTU 109

Figure 5-8: Distribution of interdeparture time for control RTUs 110

Figure 5-9: Distribution of packet sizes for control RTUs 111

Figure 5-10: Experiment network - Scenario 1 114

Figure 5-11: Experiment network - Scenario 2 116

Figure 5-12: Experiment network - Scenario 3 118

Figure 5-13: Comparison of interdeparture time for infected RLS 120

Figure 5-14: Packet size comparison between infected and clean 121

Figure 5-15: Distribution of interdeparture time for infected RTUs 124

Figure 5-16: Distribution of packet size for infected RTUs 125

Figure 5-17: Distribution of interdeparture time for covert channel 127

Figure 5-18: Distribution of packet size for covert channel 128

Figure 5-19 : Distribution of packet size for channel compared to a source with noise 130

Figure 5-20: Distribution of entropy for all channel sizes and noisy source 131

Figure 5-21: Close up distribution of entropy for all channel sizes and noisy source 132

LIST OF ACRONYMS AND ABBREVIATIONS

ACK	Acknowledgement
APT	Advanced Persistent Threat
ATM	Asynchronous Transfer Mode
C&C	Command and Control
CERT	Computer Emergency Response Team
CLI	Command Line Interface
COTS	Commercial Off-the-shelf
CSIS	Center for International and Strategic Studies
DCOM	Distributed Component Object Model
DC-OPF	Direct Current Optimal Power Flow
DMS	Distribution Management System
DNP3	Distributed Network Protocol version 3
DNS	Domain Name System
EMS	Energy Management System
FEP	Front End Processor
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
I/O	Input/Ouput
ICS	Industrial Control Systems
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineer

IP	Internet Protocol
ISIP	Industrial Security Incident Database
IT	Information Technology
KΩ	Kilo Ohm
K-S	Kolmogorov-Smirnov
KV	Kilo Volt
LAN	Local Area Network
LANMAN	Local Area Network Manager
LSB	Least Significant Bit
ms	Milisecond
MTU	Master Terminal Unit
MVAr	Mega Volt-Ampere Reactive
MW	Mega Watt
MW·h	Mega Watt Hour
NATO	North Atlantic Treaty Organisation
NEITC	National Energy Infrastructure Test Center
NIDS	Network-based Intrusion Detection System
NP	Non Polynomial
NRCan	Natural Resources Canada
NTLM	NT Local Area Network Manager
OPF	Optimal Power Flow
OS	Operating System
p.u.	Per unit
PC	Personal Computer

PID	Proportional-Integral-Derivative
PLC	Programmable Logic Controller
RAM	Random Access Memory
RLS	RTU Load Simulator
RTU	Remote Terminal Unit
s	Seconds
SCADA	Supervisory Control and Data Acquisition
SET	Social Engineering Toolkit
SYN	Synchronize
TCP	Transmission Control Protocol
U.K.	United Kingdom
U.S.	United States
UDP	User Datagram Protocol
USB	Universal Serial Bus
VLAN	Virtual Local Area Network
VM	Virtual Machine
WAN	Wide Area Network

CHAPTER 1 INTRODUCTION

Humans have always turned to technology to help satisfy their basic needs. The civilizations of antiquity built stone aqueducts to deliver drinkable water to their cities. These kinds of public works, required to support life, is called life support networks or, more commonly, critical infrastructure. As time went on and civilizations evolved, more critical infrastructure was needed to sustain human activity, economic activity in particular. One such infrastructure is the power grid. Without electrical power, most modern technology used for large swaths of human activity from entertainment to communication and medicine would cease to function. In Canada, the impact is even more direct because of the prevalence of electric heating to stave off harsh winters. So, the continued operation of critical infrastructure, and of the electric grid in particular, is a necessary requirement of modern life.

In recent years, a new threat to this continued operation has surfaced. In order to save costs, most utility operators have embraced industrial automation technology, supplied from Industrial Control Systems (ICS). This technology enables the remote operation of equipment used in the field. In the electric grid, the ICS uses a Supervisory Control and Data Acquisition, or SCADA, network to allow the automated operation of the grid. The SCADA network allows power utilities to gather measurements on the state of the grid and send commands to active equipment to alter the power flow. For example, when it was once necessary to send a technician in a truck to operate a breaker, the same operation can be made using a computer in the corporate office. However, this introduced critical infrastructure network to computer security threats.

In particular, most operators, usually profit oriented businesses, moved away from dedicated telecommunication lines to reap the cost savings benefit of packet switching networks, notably the Internet. This pushed manufacturers of SCADA equipment to converge on TCP/IP as the protocols of choice for communication. Unfortunately, the wealth of knowledge for attacking TCP/IP networks has now become transferable to attacking the critical infrastructure. This creates a serious network security risk to the reliability of critical infrastructure, such as the power grid, through the exposure of their SCADA network.

In order to address this risk, it is not possible to blindly apply traditional network security methods. Because any disruption of the SCADA system may cause unforeseen impact on the critical infrastructure controlled, care must be used when applying security methods. Additionally, SCADA networks suffer from various idiosyncrasies, such as a low tolerance for latency, that makes the use of some defensive technologies, like encryption, more complex. So, a deliberate study of how network security can be implemented in this application domain is necessary to tackle the risk.

The research project presented in this thesis strives to reduce the network security risks to the power grid's SCADA network. In particular, improvements in the ability of SCADA operators to deal with the threats of attacks from adversarial nation states is the focal point of the work. In this process, contributions to the fields of advanced persistent threat[1] strategy, experimental methods in SCADA network security and anomaly detection for SCADA networks are presented.

To get to that point, we start by providing a brief overview of the current state of SCADA security in section 1.1. This overview will reveal the high vulnerability of currently deployed SCADA systems and will analyze the current trajectory of policy efforts to tackle the problem. Then, a number of incidents involving SCADA networks are presented as a testament to the poor performance in terms of network security of current operators and a special focus will be placed on attacks by adversaries affiliated with nation states.

Using the analysis of the current situation as a starting point, we define our research problem in section 1.2. We start by analyzing the gap between the current situation and our goal. Based on that gap, we then focus the aim of our research on the goal of understanding advanced persistent threat behaviour to devise tools to defeat them and test those tools using experimental methods. In order to achieve the aim, section 1.2 also presents detailed research objectives we can use as stepping stones.

Finally, section 1.3 details the organisation of this thesis in which the efforts to achieve our research aim are summarized. This provides a roadmap to the reader of the path we used and the stepping stones necessary to attain our goal.

[1] Advanced Persistent Threats (APT) actors are threat actors which posses a high degree of skill, usually obtained through years of training and practice, and a high motivation to attack a specific target making them likely to attack the network until successful and to maintain the presence in the network once a successful attack has been launched

1.1 Current state of SCADA network security

Most security practitioners believe that no network can be 100% secure. However, knowing how close a network is to that mark can help evaluate its general security posture. In this section, we will look at the state of SCADA network security. We will start by looking at the technical vulnerability of SCADA systems, then we will review the policy efforts that have attempted to address the issue and we will conclude by looking at cyber incidents involving SCADA networks or companies in the energy sector.

1.1.1 Vulnerability of SCADA systems

According to the security firm Riptech [1], in 2001, serious misconceptions were at the heart of SCADA vulnerability. First, people would presume that the SCADA network is located on a separated, stand-alone network. Second, that strong access controls protect any access to the SCADA network. Finally, that specialized knowledge of SCADA was required to hack SCADA systems. Maynor and Graham have made similar observations at BlackHat in 2006 [2]. Within those five years, industry mentality had not progressed much while the technology supporting SCADA networks was undergoing transformation and progressing toward even more open configurations.

While most infrastructure operators are reluctant to discuss security incidents that have occurred in their infrastructure, documented cyber-security incidents do exist to testify to the existence of the problem. In one example, U.S. officials claimed that the Brazilian electrical grid, in a country known for its active cyber-gangs, was penetrated for extortion [3]. The well documented Maroochy incident where a disgruntled insider dumped thousands of litres of sewage in drinking water [4] is another example. Allsop [5] also who claims in his book to have infiltrated the SCADA system of major U.K. operator. Finally, the tale of a professional penetration tester who claimed that hacking a nuclear power plant was the easiest engagement he had participated in [6]. All of these stories testify to both the presence of exploitable vulnerabilities in SCADA networks and to the capacity of causing damage to the population by exploiting these vulnerabilities.

It could be possible that the stories that are reported are caused by bad apples with little concern for public safety. However, even Hydro-Québec, an operator rightfully regarded as having high reliability standards, suffered an incident. On December 15th 2009, an automated protection

mechanism in the Albanel center was triggered causing an outage affecting over 200,000 customers over the province of Quebec [7]. Two days later, we would learn that the outage was caused by an employee accidentally triggering the automated protection mechanisms while giving a training session in Hydro-Quebec's Rouyn-Noranda's offices [8]. Because it was caused by a human error and not by a hostile source, this incident is typically not regarded as a computer security incident. Even so, we can at least deduce the following facts:

- it is possible to cause major outages with no physical access;
- the production network can be accessed from the business network;
- deliberate malicious activity could produce the same effects that were produced inadvertently.

The last two facts strongly echo the first and third misconceptions enunciated by Riptech. In that light, we believe that the problem of SCADA systems security is current and relevant, even for residents of Canada.

In general, literature about cyber security published in power systems publications is usually well behind the state of the art in terms of computer network operations. As such, their evaluation of their vulnerability is generally overly optimistic and their evaluation of offensive capabilities is unrealistically pessimistic. For example, in the vulnerability assessment methodology for a SCADA paper from 2007 [9], Ten, Liu and Govindarasu estimate that a 7 character long password with no complexity requirement is a "good" (scores 0.33 on a scale going from 1 to 0) password policy. That kind of password policy is on par with the infamous LANMAN hashes for Windows (a pair of 7 character long passwords with no capitalization) for which any password can be cracked in seconds using widely distributed rainbow tables. As a basis for comparison, NTLMv1 (aimed at replacing LANMAN) came out with Windows NT4 sp 4 in October 1998. So, in that particular case, we are nearly a decade behind the state of the art. As recently as 2012, Nordell [10] published in the IEEE Power and Energy Magazine special publication on cyber security for electric systems a paper to promote the use of public key cryptography based on the fact that is was more secure, faster and less complex than the use of symmetric cryptography. A claim which is widely known to be unfounded.

There are numerous vulnerabilities that are plaguing SCADA systems. This situation is mostly the result of operational constraints and the attempt to leverage legacy systems in a modern environment under which they were never designed to operate.

The first operational constraint is the need to operate on deterministic delay. This makes SCADA networks particularly sensitive to denial of service (or network time manipulation in cases where timestamped messages are used). It is important to note that, even if SCADA protocols need short and predictable delays, they do not necessarily need high bandwidth because they only exchange messages infrequently. It is also important to note that a denial of service on the SCADA network usually causes a loss of control over the infrastructure and not a loss of the infrastructure. For example, if a power plant lost its SCADA system, it would lose its telemetry and the ability to remotely control the plant, yet it could still produce electricity.

The second major operational constraint is the necessity to operate without interruption. The flipside of this is that it is difficult to schedule downtime for system maintenance. It also means that any action which might jeopardize the system's uptime should be avoided. In particular, system patching, which requires downtime and occasionally leaves systems in an unpredictable state is widely avoided in the SCADA sector. In his paper, Gold [11] points out that, even with the advent of Windows 7, most utilities are still running Windows 98 equipment. This is compounded by the fact that vendors are often relying on legacy functionalities (such as anonymous DCOM for Windows which was phased out in the wake of the Blaster worm) and the vendors will rescind support if a system is migrated or patched. Even if this is not the case, utilities may well be required to go through a lengthy certification process to attach anything to a production system. The length of the process may even be orders of magnitude higher than the current patch frequency cycle (4 weeks). All of this means that SCADA networks typically run outdated software and operating systems that possess a plethora of widely known vulnerabilities. So, a hacker is usually not required to exploit (or even fully understand) SCADA protocols to gain control of a system because the underlying operating systems and supporting software are full of holes. This is contrary to the common belief among utilities operators that specialized knowledge is required to hack SCADA networks.

The third operational constraint is the need to operate without human intervention. For example, machines may need to talk to other machines without requiring a human to enter a password.

While this would not necessarily prevent the use of machine authentication, industry experts [2], [4], [12] agree that access control is lacking. Some [2], [12] even go so far as to claim SCADA systems do not make use of authentication or authorization. That observation is consistent with the use of outdated operating systems that did not provide reliable ways to perform authentication in a networked environment. In other words, any access within the SCADA network perimeter allows access to any other node within the perimeter.

The fourth operational constraint is the increased need for connectivity. The connectivity could be required for operators to interact remotely with nodes or because data needs to be extracted from the system. In fact, in his paper *The Air Gap: SCADA's Enduring Security Myth* [13], Byres says :

> *As a theory, the air gap is wonderful. In real life, it just does not work.[...] As much as we want to pretend otherwise, modern industrial control systems need a steady diet of electronic information from the outside world. Severing the network connection with an air gap simply spawns new pathways like the mobile laptop and the USB flash drive, which are more difficult to manage and just as easy to infect.*

This makes SCADA network perimeters much more permeable in reality than they are on paper. According to multiple experts [2], [4], [12] connectivity to SCADA networks is usually undocumented or thought to be non-existent. The classic example is a worm brought to the inside by a roaming laptop that is connected through a “sneaker net”. Other examples are the connection of the MTU to the corporate network to allow data warehousing of SCADA data. The mere existence of these connections is a risk because it usually allows the bridging of SCADA networks to the Internet (for example through an infected laptop). The fact that they are typically undocumented only adds to the risk, because the connections are less likely to be adequately protected.

A final operational constraint is the remoteness of the installations. This means that communications and computer equipment is often left unattended in remote locales. Both Wiles [12] and Allsop [5] testify to the lax physical security in remote substations and both claim to have physically penetrated their security in the course of a penetration testing exercise (against unnamed clients). This kind of unauthorized access can provide hackers with physical, and thus administrative, access to one (or more) RTU. In theory, this may be no worse than the damage

that can be caused if the hacker would attempt to physically damage the location collocated with the RTU. However, access to the RTU allows the hacker to have access to the complete SCADA network, which he can then leverage to achieve more widespread effects than if he damages the physical location at which he was located.

1.1.2 Policy efforts to address the vulnerability

Because of the high societal impact of cyber attacks on critical infrastructure in general and SCADA systems in particular, regulatory entities have taken steps to address the issue. We can use these efforts as a indication of the current trajectory of SCADA defense and estimate how close the industry is to solving the problem.

In the United States, where such efforts are more visible, cyber security of critical infrastructure has been recognized as a major vulnerability. A group of experts mandated by the Center for International and Strategic Studies (CSIS) argued in 2008 in their Cyber Security for the 44th Presidency report that "cyber security is now a major national security problem for the United States" [14] and Kurtz [15] recalls efforts made as early as 1996 by the President's Commission on Critical Infrastructure to address the issue. Unfortunately, it is unclear how much these initiatives have contributed to increases in cyber security.

A main axis of improvement suggested by regulatory agencies is the push for global reduction of vulnerabilities. The 2003 U.S. National Strategy to Secure Cyberspace [16] has two national priorities addressing this issue. Priority II (a national cyberspace threat and vulnerability reduction program) addresses technical vulnerabilities while Priority III (a national cyberspace security awareness and training program) addresses human vulnerabilities [16]. Assuming that the operators can correctly identify their vulnerabilities, this is still a daunting task. The underlying assumption behind the concept of generalized vulnerability reduction is that it is possible to reduce your vulnerability enough to make attacking you inefficient. It is clearly not possible to reduce the vulnerability over the entire attack surface. Figure 1-1 illustrates where various threat agents are located in terms of motivation and skill in the industrial control market.

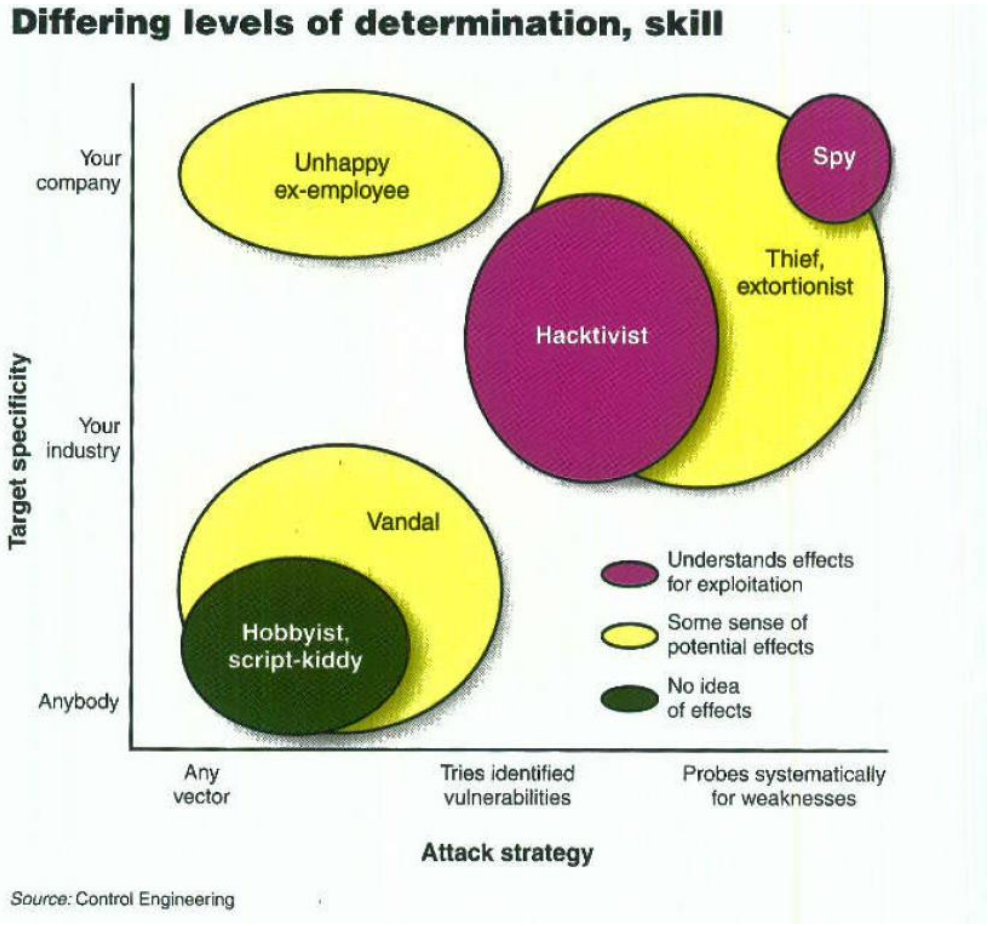

Figure 1-1 : Skill and motivation of threat agents in industrial control (© CFE media used with permission from [17])

As skill and motivation increase, it becomes increasingly costly to reduce vulnerability to a point where no risk exists. In that light, the implied objective of national vulnerability reduction programs is to address the lower left quadrant of Figure 1-1, i.e. widely known vulnerabilities affecting your industry in general. As we will see in section 1.1.3, there are highly motivated and skilled attackers, also dubbed advanced persistent threats due to their high skill level, tendency to establish a persistent presence on targets and a tendency to be stubborn in the face of active defence, that specifically target the energy sector which fall outside the scope of vulnerability reduction efforts.

This problem is compounded by the fact that the majority of SCADA operators are privately owned utilities, or publicly owned utilities that compete with the private sector. For these utilities, increasing cyber security is not a revenue generating investment. We could assume that these costs would be ultimately transferred to the customers. So, as long as no incident occurs,

this would actually harm the competiveness of a firm which would spend more in cyber security in comparison to its peers. In their paper, Dynes et al. [18] argue that, because the majority of firms have not experienced a cyber event with significant external costs, firms tend to invest only to a level that is rational, based on internal costs such as the time required to rebuild systems and lost production. The societal costs are not considered because the firms themselves do not usually bear the cost of incidents. It is therefore not rational to reduce the vulnerability to a level that would be adequate to consider national vulnerability reduction programs effective against attackers with the resources of a typical intelligence agency for example. In that sense, it is unlikely that pursuing this path will yield significant results against persistent threat actors.

1.1.3 SCADA related incidents

The most telling sign of the vulnerability of a system is the number of incidents associated with that system. However, many SCADA operators are reluctant to disclose information about breaches in their systems. Henry in [12] reports that "only 14 of the 200 Fortune 500 companies that are recognized as part of our [United States] national infrastructure actively report SCADA incidents". Of those that actively report, we cannot know if they report every incident. Even if the companies did, they can only report incidents that they have detected. Even when the incidents are reported, they are generally not distributed in the public domain. The British Columbia Institute of Information Technology Industrial Security Incident Database (ISID), which was the only open source of information of cyber incidents affecting SCADA systems, became a subscription-based product when the ISID program was discontinued in 2006 [19]. Security alerts and incident reporting from both the Canadian Cyber Incident Response Center [20] and the Department of Homeland Security Industrial Control Systems Cyber Emergency Response Team [21] are reserved for selected partners and classified "For Official Use Only" to prevent widespread dissemination. We must extrapolate from the incidents that do exist in the literature to draw conclusions.

The Maroochy water plant incident [4], where a disgruntled insider abused the SCADA system to get back at his employer, provides valuable insight on real systems. In particular, the release of the information in the public domain allowed other researchers to draw conclusions from the data and learn from the experience without needing to suffer an incident themselves. Of those lessons learned, the most telling is the high level of susceptibility of SCADA systems in general, and of

SCADA endpoints in particular, to attacks. It was possible for a terminated contractor to dump sewage water in the stormwater drains and ultimately in the region's waterways. While an unemployed worker might have a lot of time in his hands and might be suitably motivated, he usually has access to few resources. With that line of reasoning, people asked themselves what kind of havoc could an entity with greater resources and motivation wreak on SCADA systems.

The Stuxnet worm found by Symantec [22] made the world realize that nation states were interested in hacking SCADA systems for more than academic interests. They found that the worm, which remained active and undetected for many years, was specifically designed to target a specific brand of PLCs, and the software used to perform design and engineering for those PLCs. It also contained the first rootkit (a tool designed to hide the malware from the operating systems and from analysis tools) designed to work with a PLC. In addition, it contained a stolen code signing certificate, a number of zero day exploits and a sophisticated command and control scheme that allowed it to bridge the "air gap" of isolated systems. After more study, it was found that the malware was designed to cause damage to a specific type of physical equipment used in the process for uranium enrichment. Ultimately, the New York Times revealed that Stuxnet was, in fact, a cyber weapon designed by the United States to sabotage the nuclear program in Iran [23].

As revealed in the Symantec report [22], Stuxnet was probably introduced by an infected USB stick. Once a machine was infected, the malware would look for engineering files from the Step 7 program designed to interact with the targeted PLCs. It would subvert these programs to be able to spread to PLCs when they would be plugged in with a serial cable for maintenance. In addition, the malware would spread laterally on the LANs with the use of network software vulnerabilities and with USB keys. The worm also establishes a peer-to-peer network to allow the malware to update itself. Whenever a new version of the cyber weapon would be inserted, the peer-to-peer command and control network made sure the newest version of the weapon was pushed onto all the infected machines. Machines with access to the Internet would also connect through a steganographic HTTP channel to command and control servers that allowed the malware operators to push updates from outside. Tofino Security presents a good summary of Stuxnet's communications in [24] which is reproduced in Figure 1-2.

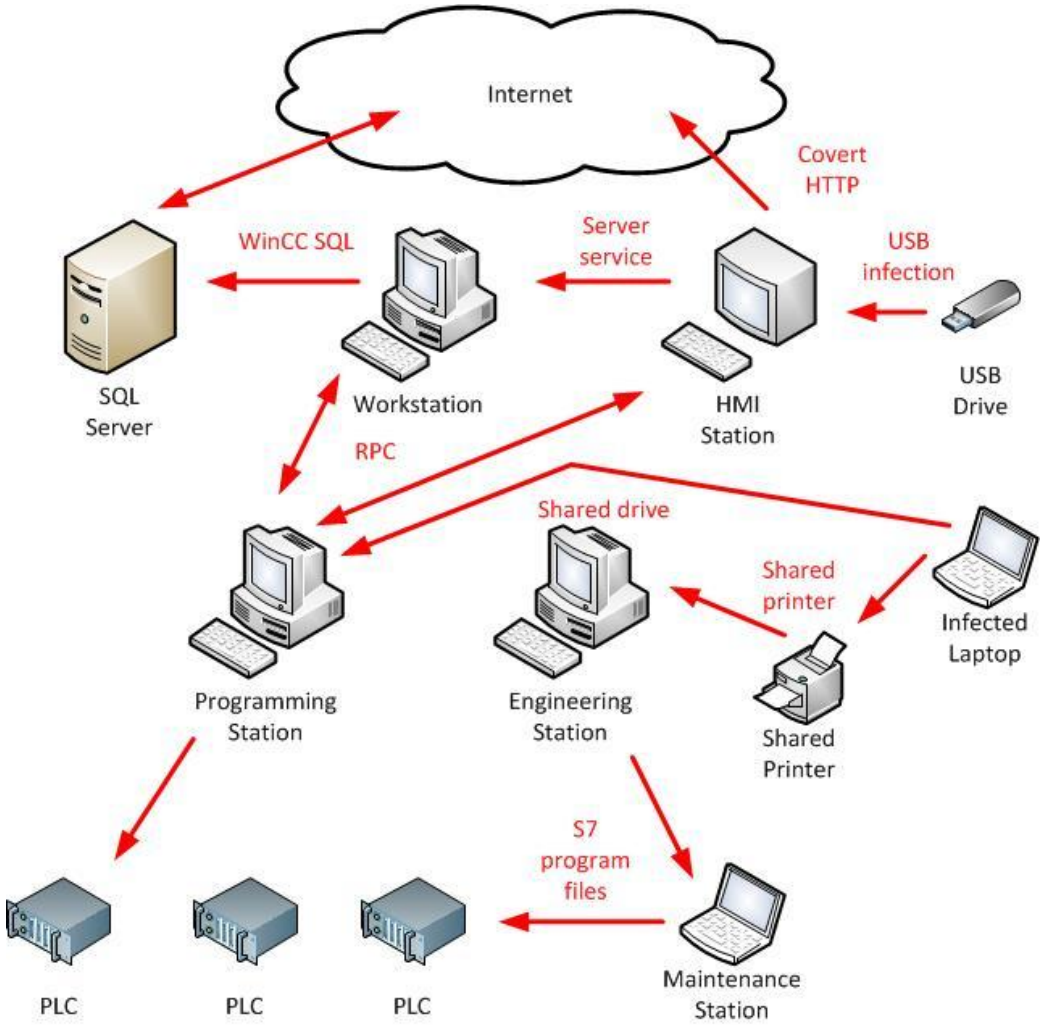

Figure 1-2: Stuxnet communications (data from [24])

Once Stuxnet was installed on a PLC, it would check if the PLC controlled a specific type of equipment. The targeted equipment was a frequency regulator for spinning motors designed for uranium enrichment and used by the Iranian government. Stuxnet would record the values sent to the engineering station for a long duration of time in order to build a model of what kind of values would be considered "normal" by operators. After this recoding period, it would start to modify the frequency of the centrifuges to make them spin very fast or very slow. This alternating would eventually prematurely damage the centrifuges, which are hard to acquire in Iran because of economic sanctions, and prevent the creation of weapons grade uranium. While doing so, it would use its recording to send operators reports that the machine was operating as normal. If a technician would connect to the machine to perform a diagnosis, Stuxnet would use its PLC rootkit functionality to mislead the technician into thinking nothing was wrong.

The discovery of Stuxnet spurred interest in SCADA security research and presented a massive wake up call for SCADA network operators. Unfortunately, due to the relatively recent discovery of Stuxnet, the efforts of researchers to create defenses is only starting to bear fruit. In a sector not recognized for its speedy adoption of new technology, it may take even longer to see a widespread adoption in the industry. On the other hand, the attackers have been cued to the vulnerability of SCADA systems. In particular, the poor state of security development of SCADA equipment and software. This provides offensive security researchers an entire field of low-hanging-fruit. Positive Technologies Security [25] tracked the number of vulnerabilities in SCADA products disclosed on public forums and we reproduce their findings in Figure 1-3. We can see that the number of reports exploded after 2010, the year Stuxnet was discovered.

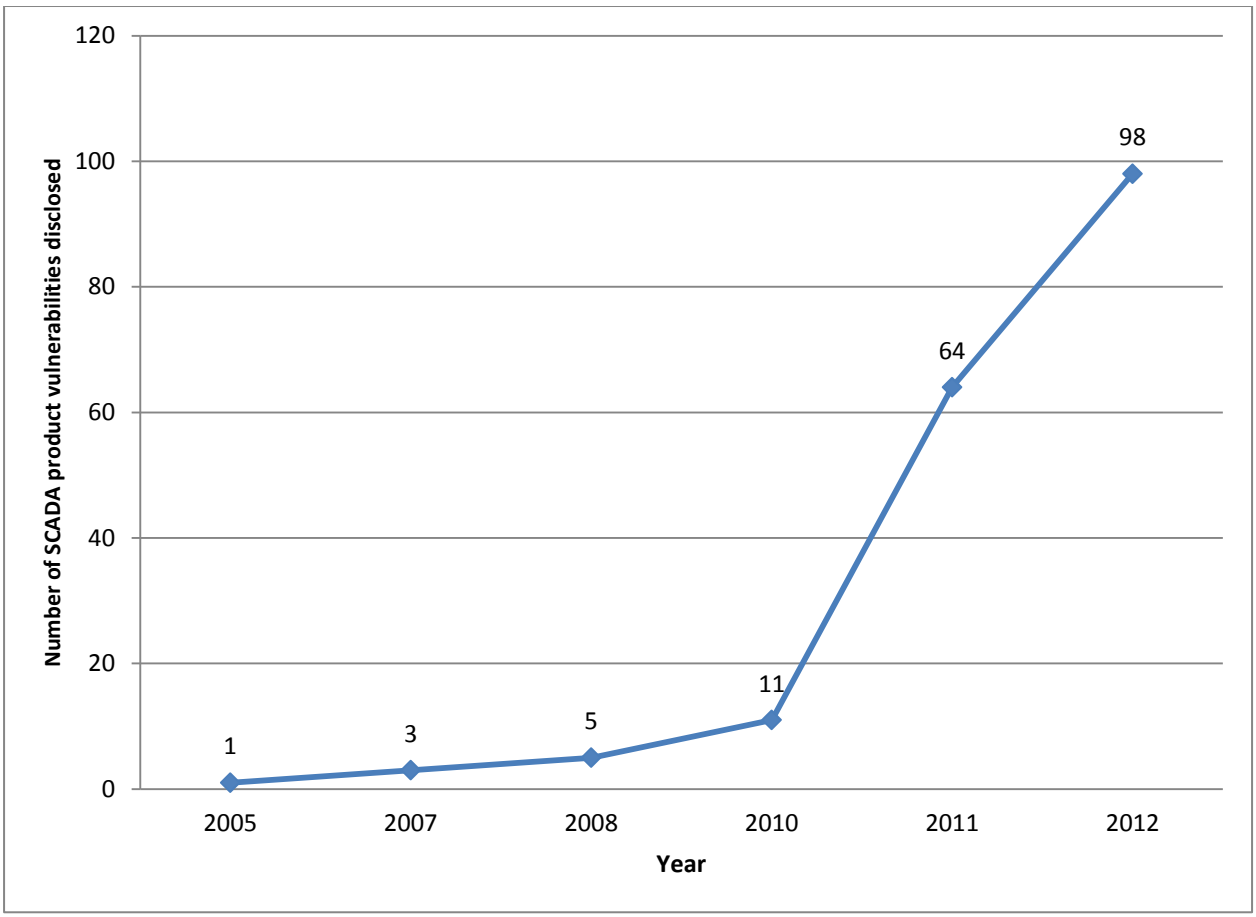

Figure 1-3 : Number of disclosed SCADA product vulnerabilities (Data from [25])

Another impact of Stuxnet is the realization that cyber attacks could cause physical damage. This information was available in 2007 when the Department of Homeland Security performed the AURORA test [26]. In that test, a power generator is made to buck wildly, produce smoke then

catch fire by sending it fake commands. However, at that time, there was much skepticism in the industry who believed the lab experiment did not accurately reflect how "real networks" were operated. This doubt was expelled by the actual damage Stuxnet did in the wild, even if it was not on power grid equipment. Because Stuxnet's code is available to everyone, it can serve as a blueprint to create attacks on other kinds of systems. To that effect, in their program 60 Minutes [27], CBS revealed that more research was done in the United States to create physical effects with cyber systems.

While Stuxnet was a very precise cyber-weapon, the Shamoon virus is more of a blunt instrument. In their description of Shamoon [28], Symantec lists multiple destructive capabilities, although the capabilities stay confined in the cyber realm. Notably, the virus wipes computers by erasing the disk and rewriting the master boot record at a specified time. So, while this logic bomb is reported to target Saudi Aramco [29], it is liable to infect any Windows-based computer and wipe it. In fact, Shamoon forced the Quatari company RasGas to shut down its servers [30]. Shamoon is widely attributed to Iran as a retaliation for Stuxnet, hinting at a dangerous escalation of reprisals for cyber attacks. In fact, recent reports warn the industrial sector that it is the target of Iranian plans for cyber revenge [31]. This underlines the fact that utility companies, even if they are not directly engaged in cyber war, can become collateral damage.

The collateral damage is not stopped at national boundaries and Canadian companies have been the target of actions by nation states. As an example, Krebs reports that the Canadian company Telvent, a company that produces and distributes SCADA equipment, was targeted by cyber espionage [32]. The goal of this attack was to acquire confidential information regarding Telvent products and to possibly gain access to Telvent's customer networks through maintenance channels. Indicators in the incident suggest the attack was perpetrated by the "Comment Crew", a group that was identified in a report by Mandiant [33] as an intelligence unit of the People's Liberation Army in China. More recent reports of spear fishing attacks[2] for the purpose of espionage targeting the U.S. energy sector to collect password and steal diagrams and plans [34] are also attributed to the same group. This kind of spear fishing attack is very effective on control room employees, with a reported 26% success rate when tested by industry researchers [35]. This

[2] A spear phishing attack is a form of phishing attack, i.e. an attack where a fake message is sent to a recipient in order to compel him to perform an action that would cause him harm, where the target of the attack is carefully selected and the message content is customized for that particular target. An example would be a message from his direct superior asking him to review a document to entice the victim into opening a document containing exploits.

suggests there is a dedicated campaign targeting the energy sector in which Canadian companies can be victimized.

In summary, SCADA systems suffer from a number of vulnerabilities stemming from operational constraints. These constraints also severely limit the capacity to actively fix the problem and policy efforts are unlikely to yield short term results. In addition, there are highly motivated attackers that have specifically targeted the energy sector to perform acts of espionage or sabotage, which generates a high risk to this sector of the critical infrastructure. The current policy, pursuing generalized vulnerability reduction is targeted at the hobbyist and the script-kiddies instead of at the highly skilled and motivated attackers such as nation state sponsored groups or large criminal gangs. In order to reduce that risk, we have to find new ways to secure SCADA systems.

1.2 Problem definition

A lot of work still needs to be done to secure SCADA networks and no single solution can solve the entire problem. This section presents how we expect to contribute to the advancement of this problem with our research. First, we present our general research goal of securing SCADA networks against targeted attacks by advanced persistent threats and we present the current deficiencies preventing the easy achievement of this goal. Then, we state our specific research aim that will advance our goal. Finally, we present the various research objectives that will be used as stepping stones to attain our aim.

1.2.1 Research goal

Incidents such as Stuxnet, Shamoon and Telvent tell us that state sponsored cyber attacks on critical infrastructure are now a fact of life. As such, it would be prudent for operators of critical infrastructure networks to take the necessary precautions to defend against these attacks. Currently, these utility operators are ill equipped to deal with this task. So, our goal is to help utility operators, in particular operators of electric grids, to defend their SCADA network against advanced persistent threats such as nation state sponsored cyber attackers.

Unfortunately for these operators, there is little publicly available information about the nature of cyber attacks from nation states, or other advanced persistent threats such as organized criminal gangs. Very recently, information, like the Mandiant report [33], has started to trickle out about

techniques, but fully understanding how these techniques are used in operations still requires expert knowledge obtained from continuous study of the field. Even with this information, it is unclear how the description of the techniques used for industrial espionage translates to critical infrastructure SCADA networks which have little value in terms of espionage. Without this information, it is hard for those operators to devise a defensive strategy.

In the absence of a coherent strategy, the focus has been put on vulnerability reduction as evidenced by the major policy efforts in that direction. While these efforts are worthwhile in the face of the apparent vulnerability of SCADA networks, they are not sufficient in that they are aimed at threat agents that do not systematically search for vulnerabilities. These untargeted threat agents are easily discouraged by any increase in difficulty, especially if they can find easier prey elsewhere. However, those threat agents that are deliberately targeting the network will be more persistent in their efforts and are unlikely to be deterred by a decrease in exposure, unless the exposure is reduced to a level where attacks become unfeasible. Judging only from the current state of vulnerability in SCADA networks, this level of vulnerability reduction is unlikely to happen in the near future. Additionally, SCADA networks are in the hands of private companies which need to turn a profit. The amount of money they can invest in defense is dwarfed by the resources available to some intelligence agencies. So, reducing vulnerability to a suitable level to prevent a nation state actor from getting in is probably not economical. Unfortunately, vulnerability reduction seems to be the only defensive strategy considered to defend critical infrastructure.

Unless they want to put their production network at risk, SCADA network operators have great difficulties in testing defensive strategies. Due to the complex nature of cyber-physical systems, i.e. computer networks where some components interface with physical devices rather than human users, such as SCADA networks, there is little publicly available data on which to perform research. Information from real deployments are typically held back because of security concerns and the financial and manpower cost of standing up a truecyber-physical experimental SCADA network at a reasonable scale is prohibitive for most researchers. As such, there is no good way to derive evidence-based conclusions about the effectiveness of defensive techniques. Until this problem is addressed, we must mainly rely on innuendos from people with access to confidential data from production systems.

In summary, if we want to target the problem of defending a SCADA network controlling the power grid against advanced persistent threats, such as nation state actors, we have to tackle a number of deficiencies in the current state of the art. Notably :

- A lack of understanding of how advanced persistent threats attacks would unfold in a SCADA network;
- A defensive posture overly reliant on vulnerability reduction which seems to be aligned for defending against untargeted threats such as commodity malware and hobbyist hackers;
- A tendency not to inform decisions from evidence-based conclusions because of a lack of publicly available experimental infrastructure.

The combination of these deficiencies makes the tackling of the problem of defending SCADA networks against advanced attackers a hard problem.

1.2.2 Research aim

Our ultimate research goal is to help power grid operators defend their SCADA networks against advanced persistent threats. To achieve that goal we aim to better understand how the behaviour of advanced persistent threats will manifest itself in a SCADA network and to develop, based on evidence derived from experiments, new tools and techniques to defeat the expected behaviour.

Our first goal is to better understand the behaviour of advanced persistent threats and how it will manifest itself in a SCADA network. By studying this behaviour we will be able to get a better understanding of the strategic goals such attackers are pursuing in SCADA network. From the strategic goals, we will be able to find constraints on attacker behaviour and devise a defensive strategy targeting those constraints. In our case, this study will lead us to postulate that the goal of the attackers is to introduce disruptions in critical infrastructure without triggering a conventional escalation. This strategic goal requires stealth so the defensive strategy should be to deny them easy access to stealth by increasing the capability for surveillance.

To create evidence of the effectiveness of the proposed strategy, we require a novel method for generating experimental data. This new method is required to be able to produce high fidelity data to test our defensive strategy on an academic budget. Additionally, this data should be able to be published in an open domain with little confidentiality constraint. In our case, this means

building an experimental setup that produces high fidelity network data with which we can test the effectiveness of network surveillance as a strategy to detect advanced attackers. The use of a hybrid emulation/simulation approach will allow us to generate high fidelity emulated network data while keeping scalability high and costs low by simulating the physical side with a power flow simulator.

Our final aim is to develop new tools and techniques to defeat the expected behaviour. Since our defensive strategy is based on surveillance, this means we aim to build a detector for advanced persistent threat communications in SCADA network. Once this detector is implemented, it can be tested using the high fidelity network data sets created by our experimental setup. In our case, the surveillance technique is actually an old technique, anomaly-based intrusion detection, that was mostly discarded because judged ineffective in conventional network environments. However, SCADA networks are different from traditional IT networks and an adaptation of this tool to SCADA networks should prove effective.

Our intuition, looking at the protocols, tells us that SCADA traffic is well regimented. The master slave architecture and the choice of polling as the primary mode of communication suggest that network traffic would be generated in a deterministic manner. More importantly, the traffic should be predominantly generated by automated processes rather than by human users. Finally, SCADA systems are usually single purpose systems with purpose built hardware to perform one function. This would suggest that the wide variety of applications that are typical of HTTP traffic should not be present. Based on all these factors, our intuition infers that, unlike traffic on "traditional" corporate networks, there is a more precise definition of "normal" over which malicious traffic would stand out very plainly. In IDS technology, anomaly-based detection is used to detect cyber attacks by finding packets that deviate significantly from a baseline representation of normality. Based on our intuition, this technology would be suitable to detect intrusions. So, we will focus on this technology as a detector.

1.2.3 Research objectives

Based on our research goal, the main research objective is to build a SCADA intrusion detector that would detect communication from advanced persistent threats. However, this task requires the realization of a number of sub-objectives:

- *Develop a model of the threat*: by reviewing incidents and by extrapolating the strategic aims of actors sponsoring advanced persistent threat actors, we will be able to synthesize the behaviour of advanced persistent threat actors and develop a model for their expected behaviour in SCADA systems.
- *Review the research in cyber security of SCADA networks*: by reviewing the literature we will be able to ascertain if another research group has solved the problem of advanced persistent threats in SCADA networks. A particular close look at how other researchers tackled the problem of generating data sets for experimental research will enable us to make sure our experiment is representative enough of a real world SCADA system.
- *Develop a methodology to perform SCADA cyber security research*: because there are no generic SCADA datasets for network security, we will need to develop a methodology that will allow the generation of high-fidelity network datasets. These datasets will need to adhere to the protocol specifications and will need to include network attacks. In addition, the datasets are required to be available to other researchers that would want to do research in this space.
- *Build an apparatus to generate data*: based on the methodology developed, we will need to integrate the various existing components and any new components we are required to build into a system that will generate data. This apparatus will also need to execute live malicious code. To that end, it will need to follow all the rules to prevent the malicious code from escaping from the experimental system to the rest of the world.
- *Select traffic features to characterize traffic*: to build a detection method, we will need to identify features of network traffic that will allow us to classify traffic between normal and malicious. These features must be sufficiently indicative of the type of application generating this traffic.
- *Characterize normal traffic*: using the features selected, we will need to build distributions that represent the statistical profile of normal traffic. The impact of a number of experiment design choices will also have to be evaluated to make sure the data produced is sufficiently representative of a large cross section of real-world systems.
- *Evaluate the detection performance of an anomaly detector*: with the help of the characterization of normal traffic, we will evaluate if an anomaly detector based on the features we selected is sufficient to detect various types of attacks.

The realization of this research project will enable us to make significant contributions to the research community. The contributions include:

- Introducing the model of pinprick attacks as a likely attack scenario from nation state actors. This work was presented at the 2010 Conference of cyber conflicts (now CyCon) [36].
- Creating a methodology to perform high-risk cyber-physical experiments for industrial control systems and building an ICS sandbox for the power grid. This work was used to provide training for a number of students from the energy sector and was presented at the 2013 International Symposium for ICS & SCADA Security [37].
- Generating high-fidelity network captures of SCADA traffic with and without malicious traffic to be used by the research communities involved in computer security, SCADA or traffic analysis. These will be made available on the web.
- Proving the feasibility of anomaly detection in a SCADA network by testing anomaly detection in our sandbox. This work was submitted but has not yet attained publication.

The sum of these contributions amounts to a significant advance in the fight to improve the security of the power grid against cyber attacks.

1.3 Thesis organization

This document presents a summary of our efforts to tackle the problem of defending SCADA network against advanced persistent threats. Various sections focus on the different efforts made to tackle our research objectives.

Chapter 2 presents a review of the current state of the art. In particular, background information required to understand the problem space is presented. Also, experimental approaches from the literature are evaluated for their suitability to produce high fidelity network data. Other research efforts in the field of SCADA security are also investigated to see if they provide insight on our research problem. In particular, attention is directed to detection oriented research and on its unsuitability to the problem of advanced persistent threat detection.

Chapter 3 studies the behaviour of advanced persistent threats to arrive at the model for pinprick attacks, our first contribution to the problem of advanced persistent threats in SCADA networks. From this model, we understand that a likely goal is disruption in a way that prevents the

defender from escalating the conflict. So, this chapter also presents a strategy to defeat that goal by undermining the ability of attackers to remain stealthy by enhancing the ability to perform surveillance.

Chapter 4 presents the ICS sandbox, a contribution aimed at providing the community with a research methodology to generate high fidelity network data for network security experiments in SCADA networks. The chapter also presents validation exercises for the ICS sandbox in the form of training sessions for members of the industry and of the replication of an impact assessment experiment from the power engineering literature.

Chapter 5 presents the results of the evaluation of the performance of an anomaly-based detector for SCADA networks that enables our surveillance-based strategy. This evaluation, realized on the ICS sandbox, shows that anomaly-based detection approaches are very effective in SCADA networks because of the regular nature of the traffic. The chapter also presents the boundary to the detection approach for covert channels that are mimicking the behaviour of the electrical network.

Chapter 5 presents the general discussion of our results and contributions to show that we have achieved our research aim. This chapter also discusses the current limitations of our work and proposes avenues for future research that have been opened by our contributions.

CHAPTER 2 LITTERATURE REVIEW

The public disclosure of Stuxnet provided a collective wakeup call about the security of industrial control systems. With this wakeup call came the inevitable conclusion that these systems, which include SCADA systems controlling critical infrastructure, are vulnerable to cyber attacks and that significant work was required to secure them. Additionally, it soon became apparent that, due to a number of idiosyncrasies such as a low tolerance for latency or limited processing power at the end points, it was not possible to directly apply well understood defensive techniques. Defenses need to be tailored to SCADA systems. This spurred researchers to invest efforts to secure SCADA networks. Due to the recent nature of the Stuxnet discovery, these efforts are only starting to bear fruit and we are still far from being able to feel confident about the security of SCADA networks, especially in the face of increasingly more sophisticated and persistent attackers.

As seen previously, the security of SCADA networks, especially of SCADA networks connected to critical infrastructure such as the power grid, is worrying. But, how worried should we be? To answer this question we need to have a clearer understanding of the impact attacks on the SCADA system can have on the electric grid. For example, what can an attacker gaining administrative access to a computer in the control room access? To achieve this understanding requires knowledge about how SCADA networks are used to control the electric grid. Additionally, we need to understand how close the research community is to solving the problem of securing SCADA networks. Particularly, we need to understand how current experimental approaches do not provide an adequate framework for academic research in SCADA network security and how current research is aimed more at indiscriminate threats than at the problem of defending against advanced persistent threats.

This chapter provides an overview of prior work in the field of SCADA security. It starts by presenting background information on the control of the power grid through SCADA systems, focusing on areas which have an impact on our research. Section 2.1 presents how the power grid functions as a network. Section 2.2 reviews basic elements of control theory as they apply to controlling an electric grid. Section 2.3 introduces SCADA networks by summarizing their components and describing how they are used for control. Building on this background knowledge, we then present an overview of research in SCADA security. Section 2.4 presents the

various experimental approaches used for research and explains how these approaches are not suitable for research focusing on network security problems. Finally, section 2.5 examines the results of the research and underscores where the research is either missing, as is the case for research about the behaviour of advanced attackers, misaligned with our objectives, as is the case for research aimed at vulnerability reduction, or is providing limited results or incomplete validation, as is the case for research aimed at attack detection.

2.1 The power grid

The goal of this research is ultimately to increase the security of the power grid. As such, basic knowledge of the operation of the power grid is necessary to the presentation of the research. To do so, we will start by looking at power lines and substations, the two basic building blocks of the power grid. Then, we will present an overview of power grid operation.

2.1.1 Power line

The power lines are long pieces of wiring that can transmit electric power between two points. In that sense, if we look at the electric grid as a graph, the power lines are the edges. While the power lines are not directed edges per se, it is not possible for electrical power to flow in both directions simultaneously. Much like a river will always flow down from the point with the highest altitude, electrical power will always from the power source, usually a power plant to the power sink, usually electrical loads such as industrial and residential consumers. This may mean that the flow of power through the line can be inverted if the transmission is reconfigured through a dynamic modification of the topology.

Unlike the ideal lines used in theoretical models, real power lines cause power loss, notably through heating. The more current carried by a line, the more power it loses and the more it heats up. For long lengths of line, the loss becomes more significant. In order to alleviate this problem, grid operators usually increase the voltage, and thus reduce the current, of long-haul transport power lines. These high voltage lines are often called transport lines and, in contrast, the low voltage lines used near customers are called distribution lines.

2.1.2 Substation

The electric substation is the location where power is switched from one line to another. Often, this is coupled with a modification of the voltage, for example switching from a high voltage transport line to a lower voltage distribution line, but pure switching may also occur. In a sense, if the lines are the edges of the power grid's graph, the substations are the nodes. Because of their ability to change the voltage level of the lines, they also act as the boundary between the various parts of the grids, for example the high voltage transport network and the lower voltage distribution network. In addition, because of their switching ability, substations act as the cornerstone of power routing redundancy. To perform this task, they often have line breakers that allow the grid operator to isolate a line, either to route the power elsewhere or to perform maintenance.

Figure 2-1 : Power flow through a typical substation (Reproduced from [38])

The United States Department of Labor provides us with a schematic of a typical substation and a description of how the power flows through the substation. The schematic is reproduced in Figure 2-1. The power comes from the transport network's incoming subtransmission lines with a high voltage (34 KV) and passes through a series of air-break switches and circuit breakers that act as a protection layer. Then the power "steps down" in voltage to distribution level voltage (7.2 KV) in the transformer and is relayed to the distribution bus. The power can now be switched to the various outgoing distribution lines. Cutout switches also allow lines to be isolated. In modern power grids, all this equipment is equipped with sensors and remote operation devices. The

control house is used as the server room to host all computing and telecommunications equipment that is required to perform these functions.

2.1.3 Overview of the power grid

If we look at the power grid as a graph, the core of the graph, where the inter-connexion resides, is the transport network. Because the transport network needs to cover long distances, where reliability might be an issue, and because it needs to route power from a few sources to a large number of customers, transport networks typically have a meshed topology. While the exact degree of connectivity depends on a number of factors such as cost, right of way and geography, a higher degree of connectivity is preferred because it enables more control over routing which has benefits for both load balancing and reliability.

On the other hand, the connectivity of power sources is typically more limited. Because of the often remote location of power plants, it is usually not economical to have a number of power lines connecting them to long-haul transport networks. Especially since electricity is typically not produced at transport level voltage and needs to go through a substation to connect to the transport network. Also, as previously mentioned, power needs to flow to a sink. It is therefore not useful to connect the various production sites to one another. As such, power sources typically have a substation directly on site to convert the power for a high voltage line and are connected to one, or two if the utility company wants redundancy, transport switching substations.

On the distribution side, the sheer number of customers would make it prohibitively expensive to have dedicated substations as is the case for power plants. Naturally, some big industrial or institutional customers, like aluminium production plants and hospitals, might have more dedicated facilities, but that is the exception rather the norm. So, it is typical for distribution substations to route power to a number of distribution lines going to various clients. Each of those lines act as a bus from where all the clients in the neighbourhood tap in to get their power, even though the entire neighbourhood can be summarized in a single sink. This usually creates a star topology where a number of sinks are connected to one, or more for redundancy, distribution substations. These substations are in turn connected to one, or more, transport switching substations.

In summary, the electric grid is divided into three sections: the production, the transport network and the distribution network. The production section contains all the power sources, the transport network performs the routing and delivery and the distribution network contains all the sinks. Between each zone, there are substations that convert voltage levels and allow for isolation. Figure 2-2 presents an overview of what such a grid might look like. In North America, this separation is also usually enforced through anti-monopoly regulation. The North American energy market considers that operating all three sections constitutes a vertical monopoly and is an unfair competitive advantage. This has led state monopolies to either split into multiple companies owned by a single shareholder (as is the case with Hydro-Québec Production, TransÉnergie and Hydro-Québec Distribution) or to deregulate and adopt a market-based approach (as is the case in Ontario). In terms of industrial automation, this fragmentation of the companies ensures that the control of each section of the power grid is often done independently for large utilities or in small islands containing the three sections for smaller utilities.

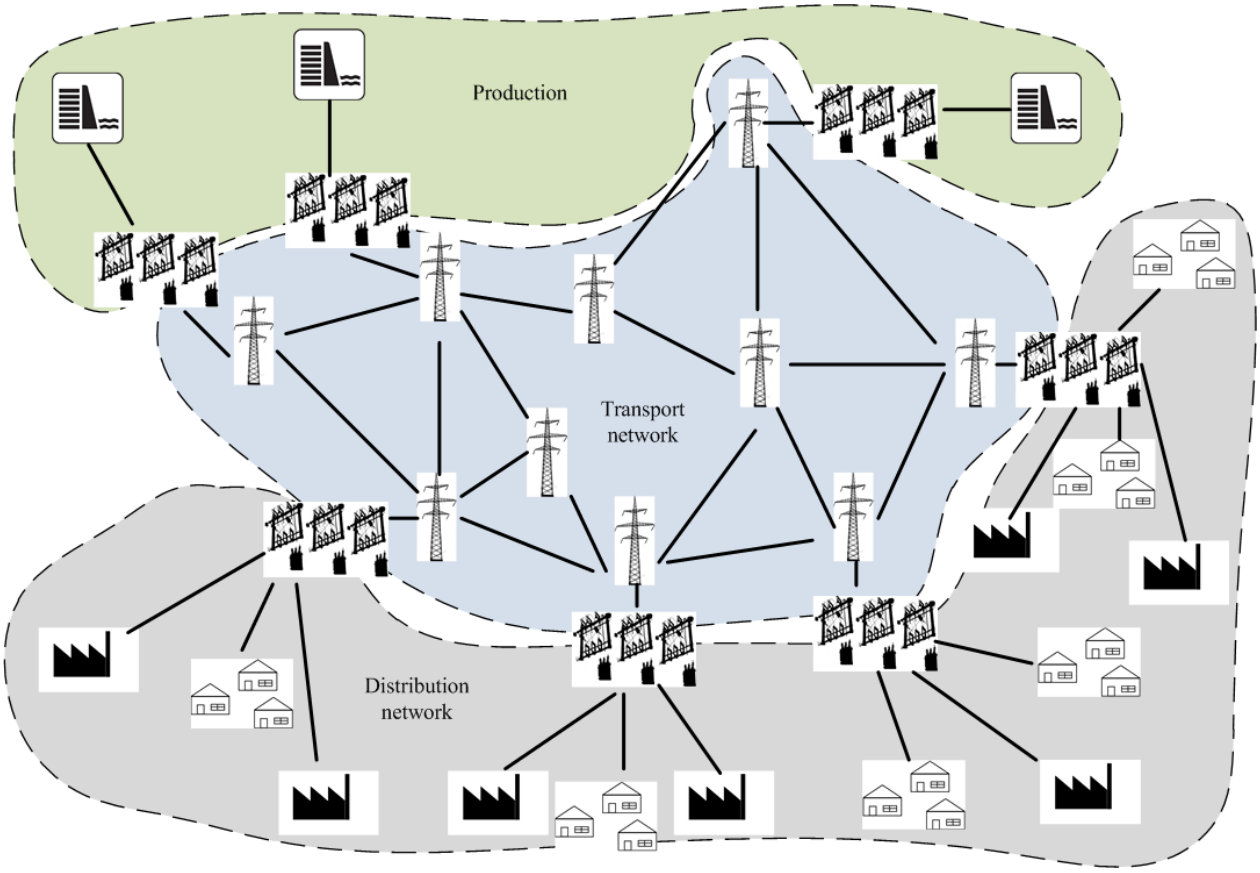

Figure 2-2 : Overview of the grid

2.2 Elements of control

Modern power grids are too large and too complex to operate manually. Utility companies require the help of automated control to ensure the smooth operation of the grid. In order to understand how industrial control systems are implemented and used to control complex systems such as power grids, we need to revisit some basic control theory concepts. We can then study how these concepts apply to the electric grid. Finally, we describe how the control center acts as the brain that adjusts the control, based on power grid operations.

2.2.1 Basic control theory

All control schemes follow a basic principle: a desired state of the system is set, a deviation between the actual and desired state is calculated and a pressure is applied to steer the state of the system toward the desired state. This principle defines the concept of the feedback loop illustrated in Figure 2-3.

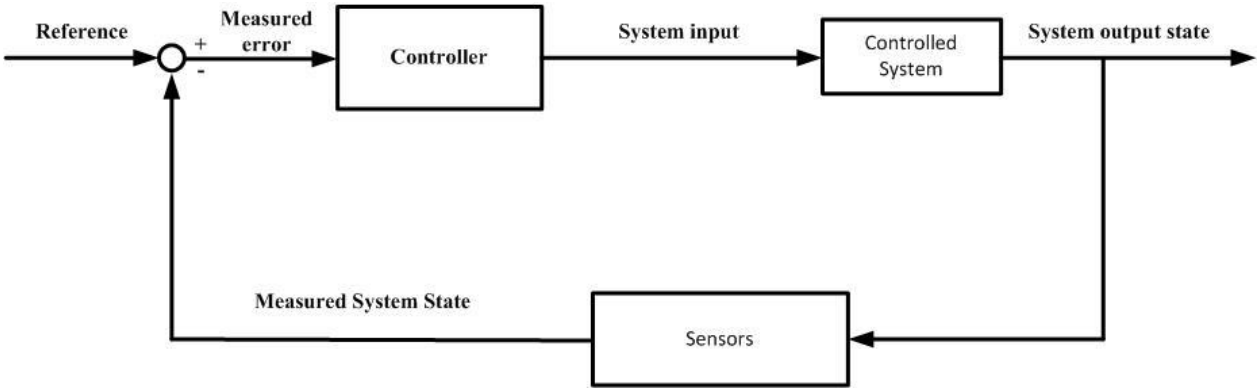

Figure 2-3 : Basic control loop

The reference is the desired state, the measured output is the actual state of the system and the measured error is the difference between the desired state and the measured state. Pressure to the system is applied by modifying the system input to guide the system output, i.e. the system state, toward the desired outcome. Once the system output is in the desired state, the measured error will be zero and the controller will stop applying pressure. This process is dynamically repeated for the entire operation of the system.

As an example, a driver on a highway wants to drive in a straight line in the middle of his lane 500 meters behind the car in front of him. That trajectory is his reference. Using his eyes as a sensor, he can gauge if his trajectory deviates from the straight line and estimate the error. The

brain, acting as the controller, will provide a series of inputs (turn the wheel, ease off on the gas pedal, etc.) that will, hopefully, guide the system toward the desired outcome. The driver will keep adjusting his distance and trajectory until satisfied with the outcome (i.e. the measured error is zero). He will then continuously monitor his situation to make sure that the state stays in the desired state even if another car brakes in front of him for example.

Even control of complex systems follows the principle of the feedback loop. However, as the system to be controlled becomes more complex, such as the electric grid, it becomes more difficult to observe and describe its state. The number of available inputs may also increase dramatically and the exact relationship between the inputs and the outputs may not be completely understood. This makes the job of both sensors and controllers more difficult and often requires multiple sensors and complex calculations by the controller to assess the correct values for the system inputs.

2.2.2 Application to the electric grid

The electric grid is a complex system with a complex state. This makes controlling the system more difficult than steering a car in the middle of the road or making an elevator stop on the correct floor. To tackle this job, the use of a Supervisory Control and Data Acquisition, or SCADA, system is required. This system is a collection of sensors and system inputs that can be used to collect data about the state of the system (i.e. data acquisition) or to modify inputs to alter the state of the system (i.e. control element). In terms of the basic control loop, the SCADA system provides both the system input and the measured output. The controller element is either provided by a human operator sitting in a control facility estimating state using his experience or pre-defined operating parameters and alarms, or by an automated Energy Management System, or EMS, that can perform automated monitoring and control tasks based on an estimation of the state of the grid. Figure 2-4 illustrates those control loops.

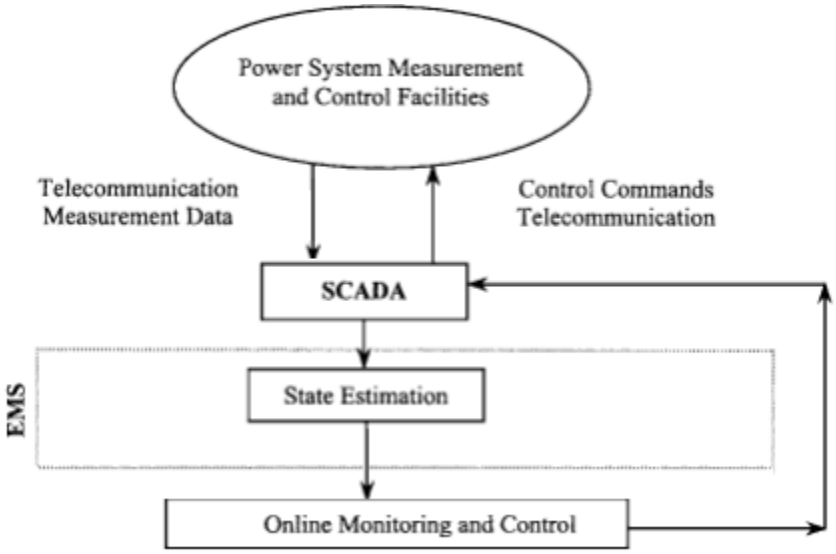

Figure 2-4 : Control loops for the electric grid (adapted from [39])

As an example, let us consider a simple reference state where power supply is equal to power demand. On a cold winter day, as the temperature drops, demand for power rises. Supervisory elements measure the state of the grid and notice that some metrics are outside normal operating parameters. These metrics are analyzed at the telecontrol center or by an automated power management system and the control logic suggests to increase power supply. A command is sent to the actuator of the intake door at a hydroelectric plant to increase the volume of water going to turbines to make them spin faster. This will produce more energy and supply will meet demand. Naturally, this impacts thousands of other states. Perhaps the turbines are now spinning too fast, maybe the high voltage line falls outside operating parameters, a transformer overheats or the drop in the water level jeopardizes profits for the third quarter. Each of these individual elements is a part of the general state of the electric grid for the basic control loop and contributes to the complexity of the control.

To solve the problem of state complexity, the divide and conquer approach is typically used. The network is partionned and a control is applied on the partitions. In their book, Shahidehpour and Wang [39] provide a system partionning approach for voltage control. For SCADA systems, the ultimate consequence is that the complexity of the global state is also managed by breaking it down in parts. Ultimately, it is possible to break the grid down to every single piece of equipment and control each individually. However, in order to have a very detailed control scheme, it is necessary to have a fine granularity of information and the ability to make complex decisions

based on a large volume of data. In addition, detailed control of the input of the system is required. By leveraging modern communication protocols and computing power, this is precisely what has been done for the electric grid. Each piece of equipment is now connected to sensors and actuators to become a SCADA termination point.

To enable SCADA systems to be used to control the electric grid, two types of points exist: measurement points for points that are sensors and control points for points that can alter the state of the system. We now have the control loop illustrated in figure 2-5.

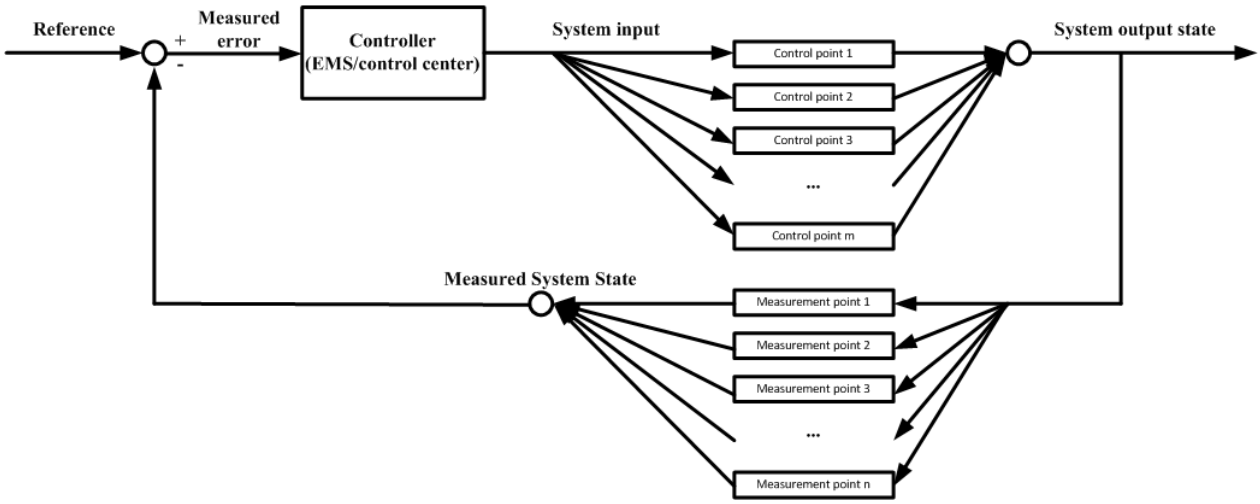

Figure 2-5 : SCADA control loop

Based on measurements by SCADA measurements points, we eastimate a state S = {$MeasurementPoint_1$, $MeasurementPoint_2$, ..., $MeasurementPoint_n$}. Based on this estimation, we calculate the error based on business objectives or operating limits. The controller then sets the values of SCADA control points to provide the system input I = {$ControlPoint_1$, $ControlPoint_2$, ..., $ControlPoint_m$}. This, in conjunction with outside factors that cannot be controlled such as customer demand, weather, physical properties of equipment and so on, define the state of the system output. The output is measured by measurement point sensors to close the loop.

2.2.3 Control center

Ultimately, all control must obey some form of control logic. This control logic requires conscious design and massive data processing. In modern systems, large parts of that logic can be fully automated with the use of automated software such as Energy Management Systems (EMS) and Distribution Management Systems (DMS). These systems can perform a number of functions

based on mathematical calculations. For example, if there is an incident and a line is suddenly taken offline, a computer will be able to calculate the correct rerouting of power to remaining lines to avoid going over operation limits far quicker than a human could. However, these automated routines cannot adapt well to unforeseen situations. Moreover, it might not be efficient to spend time to create automated responses to preplanned events that are out of the ordinary, such as isolating specific equipment for maintenance. Such functions are typically left to human operators sitting in a control center.

The main function of the control center is to maintain situational awareness of the status of the grid. This allows operators to be notified of the occurrence of failures and to be able to respond accordingly. Another important function of the control center is to support operations, for example, by manually rerouting power for economic reasons or liaising with maintenance crews to alter the grid's topology. The main tool to achieve these objectives is the Human Machine Interface, or HMI, stations provided to each operator. These stations present the operator with a graphical interface to visualize the data collected by the sensors in a coherent way. For example, a schematic of the electrical network can be created and the values of each sensor can be positioned next to each piece of equipment. In addition, the HMI provides the operator with a metaphor to perform manual control of pieces of equipment. For example, clicking on controllable equipment in the schematic might bring up a contextual menu that enables remote control. Finally, preprogrammed alarms based on predefined operating limits can help operators identify faults and locate pieces of equipment that may be responsible. The HMI may provide a general alarm browser, or locate alarms in a visual context, for example by making a piece of equipment turn red, or even provide sound notifications through speakers.

To perform all these functions, control center equipment needs to communicate to the SCADA software for both measurements and control. This is done through vendor specific proprietary HMI protocols and thus requires computer network connexions. As such, operator consoles running HMI software typically sit on the same local area network as the SCADA central server on the "production" LAN. Because the operators also need access to various enterprise services, such as Active Directory for authentication, mail for communication, and so on, the workstations also need to reside on the "office" LAN. This situation can be resolved in a number of ways depending on the utility company's risk tolerance and budget constraints. For example, one company might provide two workstations for each operator while another might just collapse

both the production and office LANs into a single intranet. A middle ground approach of dual-homed workstations, with one network interface card connected to each LAN, is also common.

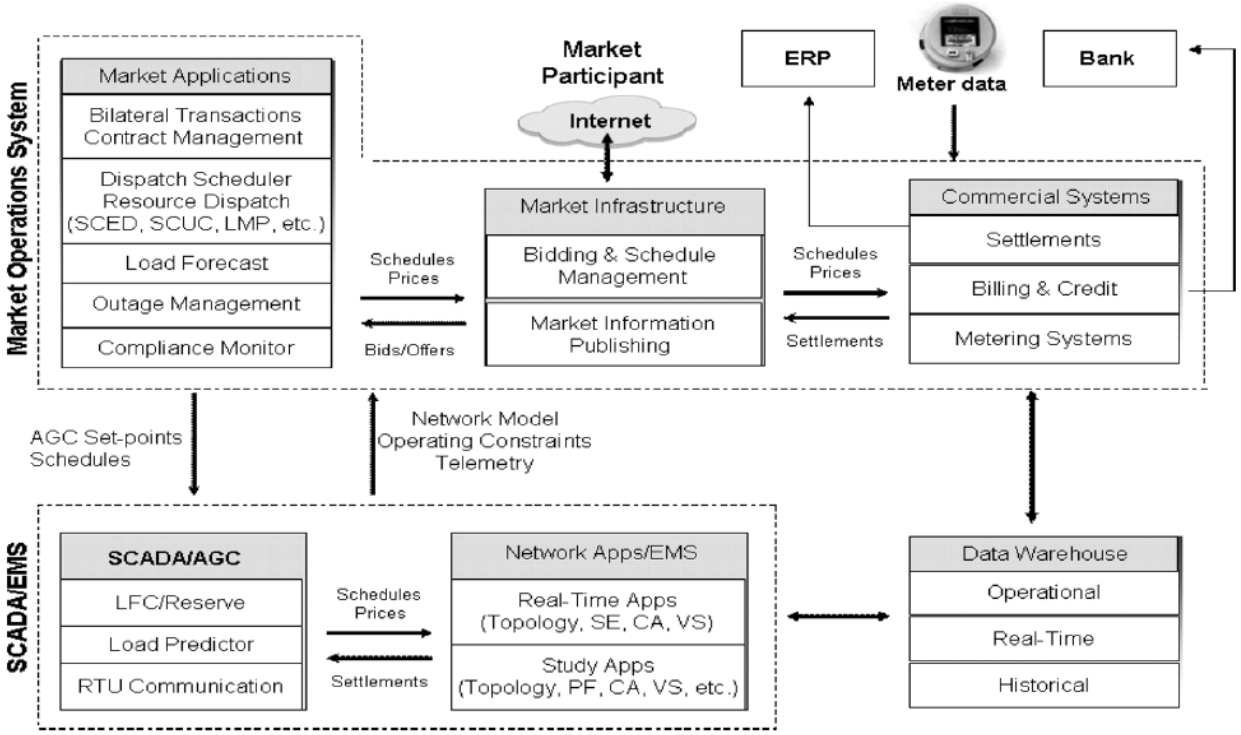

Figure 2-6 : Control Center functions (reproduced from [40] © 2005 IEEE)

In addition to these workstations, the control center typically hosts a number of business applications that require direct access to SCADA data. Figure 2-6 illustrates some of those functions. For example, a utility company might want to save all the values of the SCADA sensors in a big database, or data warehouse, in order to be able to look at historic trends. Perhaps the value of power delivered for the purpose of billing is calculated from data provided by SCADA enabled sensors. These applications are typically made available to users or process coming from inside the company. Outsiders often need to have direct access to SCADA data as well. For example, dynamic pricing for energy markets requires real-time access to a number of metrics to estimate supply and demand and establish prices. The utility company might also need access to data from other utilities or from major customers that are running their own SCADA network. For example, if the utility buys power from a privately owned power plant, it would require real-time data on what is available. For that purpose, the control center may require connexions to other control centers, including ones from external partners.

2.3 SCADA system architecture

A cornerstone of the industrial control system for the power grid is the SCADA network because it acts both as sensors and as control input. In this section, we look at the SCADA system architecture for the power grid. We start by presenting a high level view of SCADA architecture and the role of each piece of SCADA equipment. Then we look at examples of each of those pieces of equipment. Finally we cover the DNP3.0 protocol which is the protocol most commonly used in power grid applications.

2.3.1 SCADA system architecture

We have seen that SCADA networks use a distributed approach for the control of the electric grid. As such, SCADA networks build layers of automation, starting with physical devices installed on power systems equipment.

The SCADA devices connected to power systems equipment are usually hybrid analog-digital devices. They require an analog component to interface with the power system equipment and a digital component to interface with the SCADA network. The device can then either digitize analog values for measurement points or convert a digital value into a physical action for a control point. For example, a device may convert the analog value of a voltmeter into an instantaneous digital floating point value, feed an analog voltage setpoint value to a PID (proportional-integral-derivative) controller connected to an autotransformer or activate a hydraulic jack that will turn a breaker off. To perform this wide array of tasks, the various measurement and control points use Programmable Logic Controllers, or PLCs, that can be programmed to perform a range of control tasks. In that sense, PLCs are the hands and eyes of the SCADA system and they form the first level of automation.

The PLCs need to physically interface with the machines. Because of this, they are usually colocated with the power system equipment they are controlling. In terms of the power grid, this means that power system sites such as transport and distribution substations, power plants, large distribution sites, and so on host a collection of PLCs. It is convenient to enable these sites to have local control without going through the central telecontrol center. So, it is customary to aggregate the data from all the PLCs in the same physical site to a remote terminal unit (RTU). This is done by connecting the PLCs to the RTU using short range telecommunication

technologies such as Ethernet wiring, serial RS-232 wiring or wireless conenctions such as ZigBee. The RTU allows local operators to read the values of measurement points and operate control points through a local human-machine interface (HMI) station and perform local control. In addition, the RTU can convert the communication to a protocol, such as TCP/IP, that is suitable for long-distance transmission towards a central controller located in the central telecontrol center without the need for dedicated telecommunication lines for each PLC. This provides the second layer of automation.

In the case where there is only a small number of PLCs on a site, for example, next generation meters in smart grid applications, it may not be resource efficient to deploy an RTU. In these cases, an Intelligent Electronic Device, or IED, which combines the functions of a RTU and a PLC may be deployed instead. However, in terms of network architecture, it is functionally equivalent to a RTU with a very small number of PLCs connected to it. As such, throughout the text, we will only consider an architecture with only RTUs as the second layer of automation, but we keep in mind that these RTUs could be replaced by IEDs.

The master terminal unit (MTU) is connected to all RTUs within a region and aggregates the data and provides control to all these sites. As such, the MTU is typically physically located in the control centre of the electrical grid operator. HMI consoles for human operators are also typically collocated on the same network as the MTU. In most cases, this operational network is separated from the operator's administrative network by a firewall. However, for cost saving reasons, some of the operator stations might reside on both office and production networks to allow operators to read email and access the Internet on the same workstation. A historian application, i.e. a database that records all historical values of measurement points, might also reside on this operational network. This historian will typically require some communication with the office network in order for office workers to perform data analytics or to support other business functions such as billing.

Overall, the SCADA network for the control of a power grid is a logical tree network with the MTU at the root of the tree. The MTU is connected to RTUs, who can be connected to PLCs or intermediate RTUs. Finally, the PLCs are connected to either control points or measurement points. Figure 2-7 illustrates a typical SCADA architecture. In this figure, each subdivision represents a physical location such as a power substation. Each subdivision hosts one RTU

connected to the network and could also host a local HMI if local control is required. On the other side of the SCADA WAN, the MTU sits in the control centre to administer an entire region.

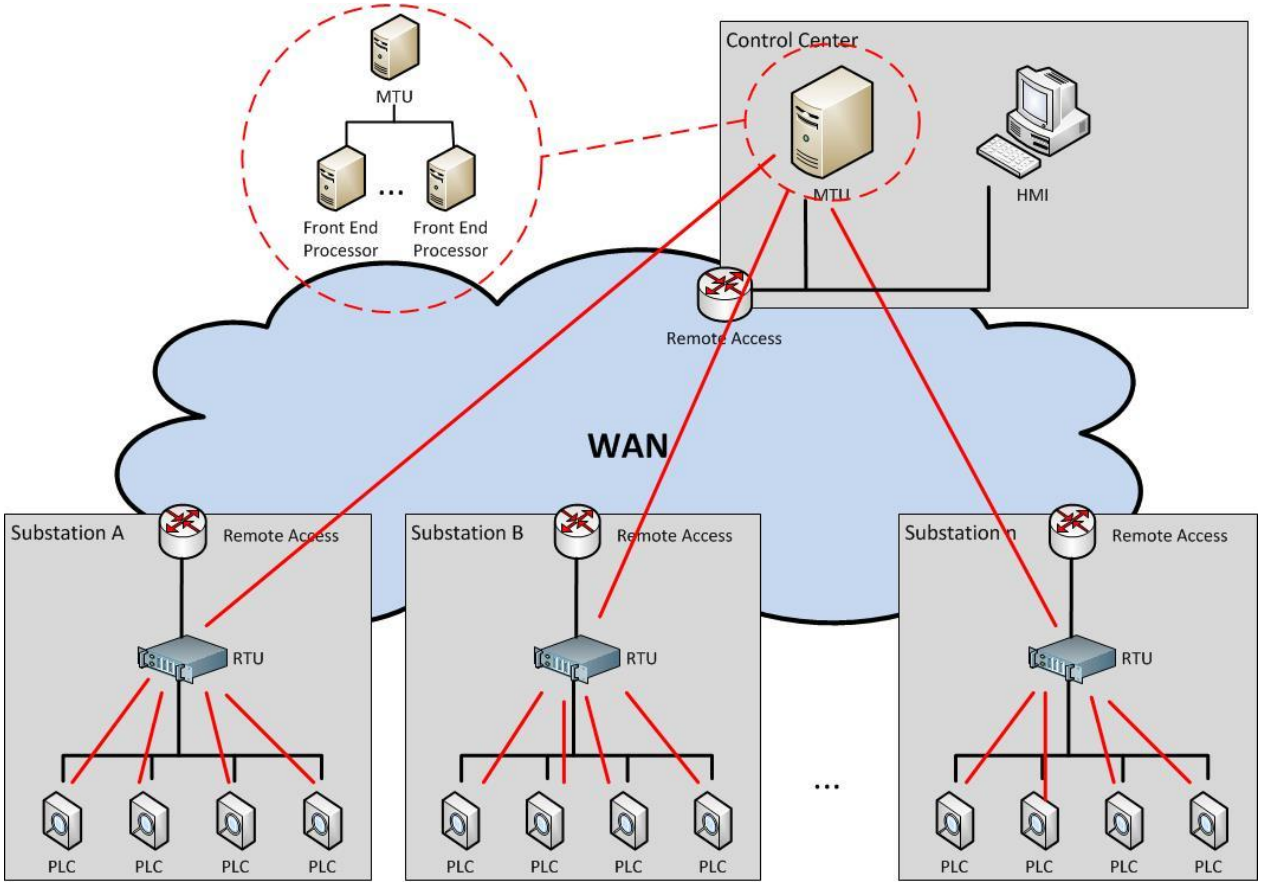

Figure 2-7 : SCADA architecture for the power grid

2.3.2 Equipment examples

While SCADA architecture seems simple enough on paper, the devil is in the implementation details. Each device operates differently based both on type of equipment and manufacturer. Devices are often custom made for one particular piece of equipment and so on. However, core functionality remains the same and most pieces of equipment of a certain type, even if one is a PC with custom I/O cards while the other is a custom-built circuit board, operate in the same way. As such, it can be useful to look at examples of SCADA equipment to better visualize the functionality each offers.

At the control center level, all the functionality is typically software built on COTS type hardware. Control center applications typically only interact with data stored in databases. They can usually rely on TCP/IP over Ethernet for all their communication needs. In that sense, an

MTU or a historian looks like any other rack mounted server. The actual number of servers required depends on the number of functionalities provided and on the scale of the network. For example, a national level utility might host their historian database on a dedicated server and place Front End Processors (FEPs) to handle the communication with the RTUs to achieve better performance.

The most important functionality offered by the MTU is to provide the data used in the HMI for the operators in the control center. Multiple kinds of metaphors and visualisations can be used to help operators place the data in context. Figure 2-8 presents an example of a bare bones HMI metaphor that presents values and offers contextual menus for control. The color coding of abnormal values is also presented as a typical visual aid to identify problems to the operators. The top left menu also shows a number of other applications of the HMI such as the alarm viewer (3rd button from the left), the networking monitor (5th from the left) and the trend graph display (9th from the left).

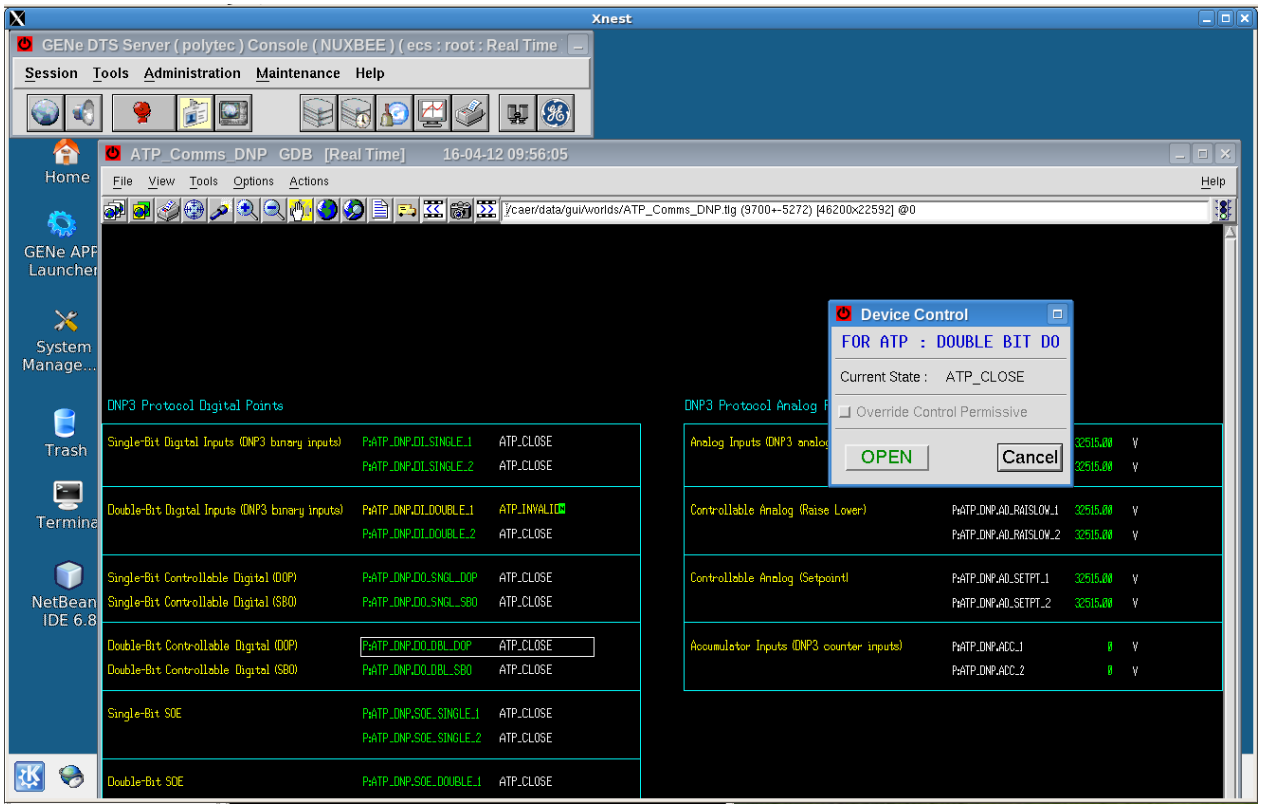

Figure 2-8: Example MTU HMI

Unlike MTUs, RTUs need to communicate with devices that may not support Ethernet. As such, RTU design is based around what type of communication it needs to offer. Each RTU has at least

one Ethernet or WAN port to communicate with the MTU and multiple ports to communicate with PLCs. Each of those ports will be designed to fit a protocol, whether it be RS-232 or Zigbee employed by that particular PLC. Figure 2-9 provides an example of a modular RTU where you can fit specialized I/O modules for each piece of equipment. In addition to the communication functions, the RTU hosts some applications to allow protocol conversion and local control amongst others. These applications may run on a variety of architectures from custom embedded software to web applications running on Windows.

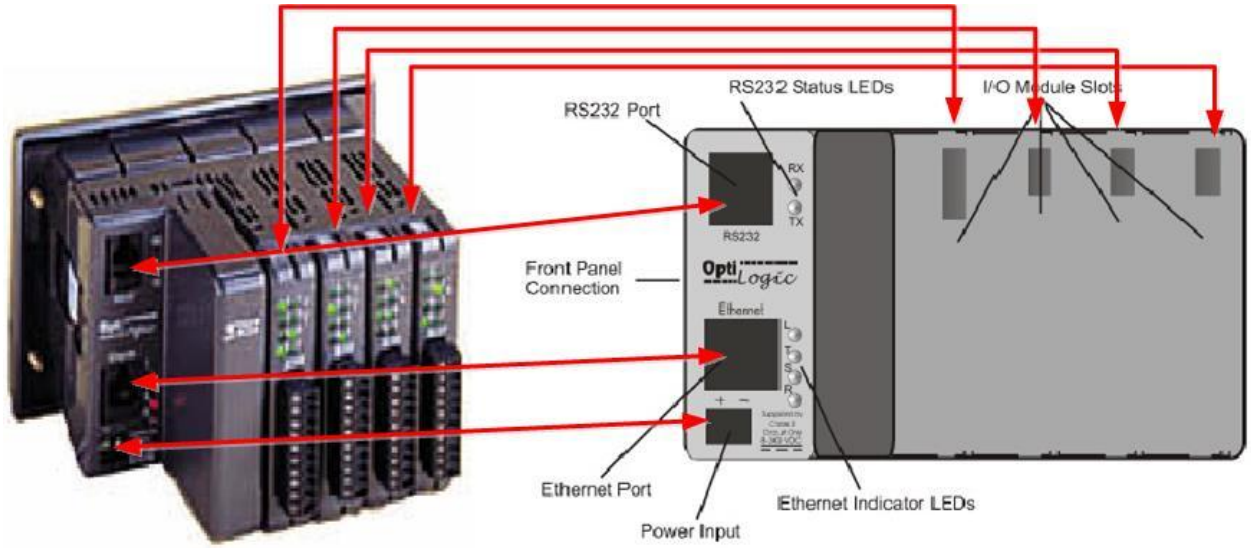

Figure 2-9: RTU example (adapted from [41])

Because PLCs are more intrinsically linked to the physical piece of equipment on the electric grid, they are one step more removed from COTS software and hardware than RTUs. As such, they are defined by the type of signal they collect or send to the physical equipment. Figure 2-10 provides a good example that illustrates that most of the bulk of the device comes from I/O cards. In terms of functionality, PLCs seldom need to provide a large number of applications because the device is designed to operate with specific pieces of equipment. As such, it does not require as much programmability and adaptability in terms of processing power as an MTU or even an RTU does. As such, they tend to focus on embedded architectures and operating systems. A study of commercially available PLCs by Schwartz and al. [42] found that ARM, Motorolla 68000 and Power architecture were preferred for architectures while VxWorks, Windows CE, QNX and the occasional Linux were prevalent for operating systems.

Figure 2-10: Example PLC (reproduced from [43])

2.3.3 DNP3 protocol

Because SCADA networks emerged from closed systems built around particular brands of industrial automatons, the SCADA networks of today communicate over a wide variety of protocols. The vast majority of protocols are proprietary protocols developed by manufacturers of SCADA equipment and are understood only by their own brand of automatons. However, during the 1990s, efforts were made to standardize SCADA protocols in order to allow interoperability between the various brands of automatons. While market forces proved stronger than the interoperability efforts, the Distributed Network Protocol version 3 (DNP3) and IEC 60870-5-101(along with ModBus from Modicon) have managed to become de facto industry standards. The main advantage of these two protocols for academic researchers is that these protocols are so-called “open protocols”, meaning that the protocol specifications are available (for a fee) on request. For our purposes, we will concentrate on the DNP3 protocol because it is the protocol of choice for North America (IEC 60570-5-101 is more popular in Europe), particularly in the electrical sector.

According to Clark and Reynders [44], DNP3 offers a large feature list including:

- Time stamped messages for sequence of event recording

- The breaking of messages into multiple frames for better error control and quicker communication sequences
- Peer-to-peer communication as well as master-slave
- Support of multiple master topology
- User definable objects
- Unsolicited (i.e. without polling by master) reporting of exceptions/events
- Support for “changed data” only response
- Broadcast messages
- Secure configuration and file transfers
- Addressing for 65 000 devices
- Time synchronization
- Acknowledgements on data link and application layers

Because of these features, the typical DNP3 mode of operation is the so-called “quiescent mode”. In that mode, there is no need to frequently poll the sub-stations in order to determine if a change occurred. The master sits “quietly” and waits for nodes to report significant changes in status by means of “unsolicited reporting”. Periodic polling is still used, but mainly to detect communication failures. The peer-to-peer communication capabilities also allow for a sort of hierarchical organisation where a substation can act as a master for other substations and relay information to the actual master.

The DNP3 protocol is loosely built on the Open Systems Interconnection (OSI) model in the sense that it possesses multiple layers performing various functions (e.g. the physical layer deals with the physical means of communication) that are encapsulated in each other. However, in part because in the early days of DNP3 all connexions were point-to-point and no routing was required, DNP3 only has four layers: physical, data link, pseudo-transport and application. The full implementation of every layer also allows DNP3 to fulfill SCADA requirements, in particular the need to process packets in deterministic time. Figure 2-11 illustrates the various layers and their encapsulation headers.

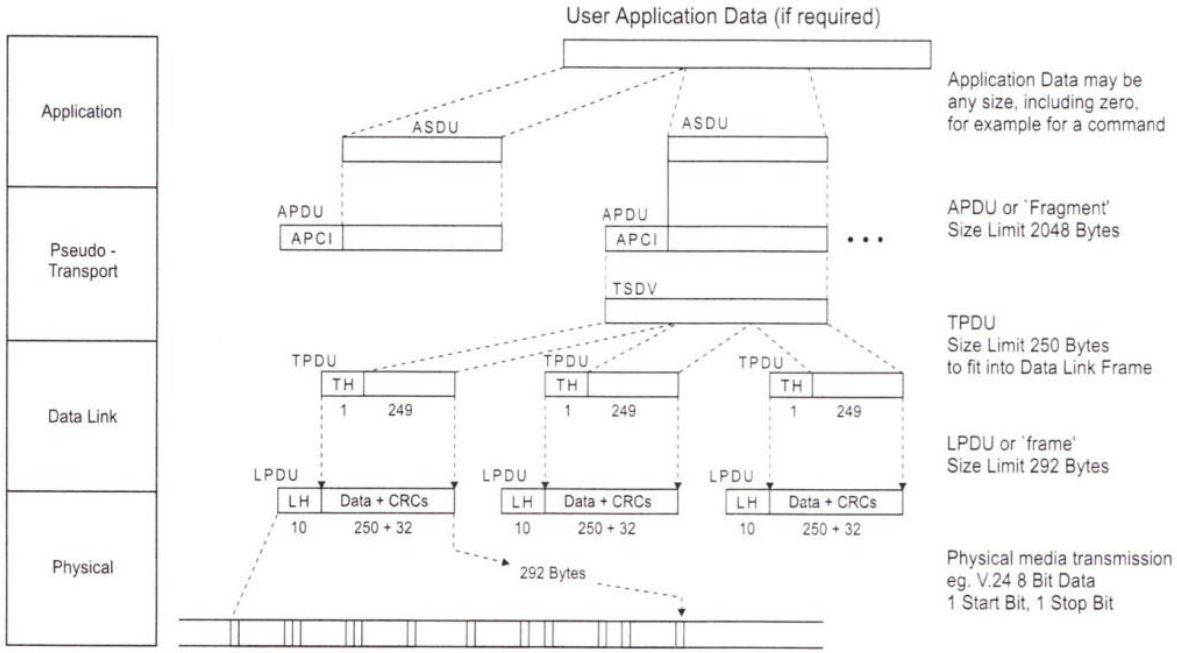

Figure 2-11: DNP3 layers (reproduced from [44])

As seen in the figure, DNP3 implements all four communication layers. This approach is different from traditional communication design where specialized and interchangeable protocols deal independently with each layer. For example, in typical Internet communication, the Ethernet protocol is the data-link protocol, the IP protocol the network protocol and TCP is the transport protocol. As far as these typical protocol architectures are concerned, the transport protocol could very well have been UDP and the data-link protocol could have been ATM. That is not the case for DNP3.

In order to modernize the protocol to make use of the developments in communications (i.e. cheap and fast communications using IP over Ethernet), it was necessary to adapt the DNP3 protocol to allow the protocol to use a non-DNP3 transport, network and link layer. This led to the creation of the DNP3-over-IP specification. The concept behind DNP3-over-IP (thereafter referred to as DNP3) is to fully encapsulate DNP3 in an Internet communication. This means that the classical DNP3 physical layer is (typically) replaced by the TCP over IP over Ethernet combination. Figure 2-12 shows a DNP3-over-IP packet is created.

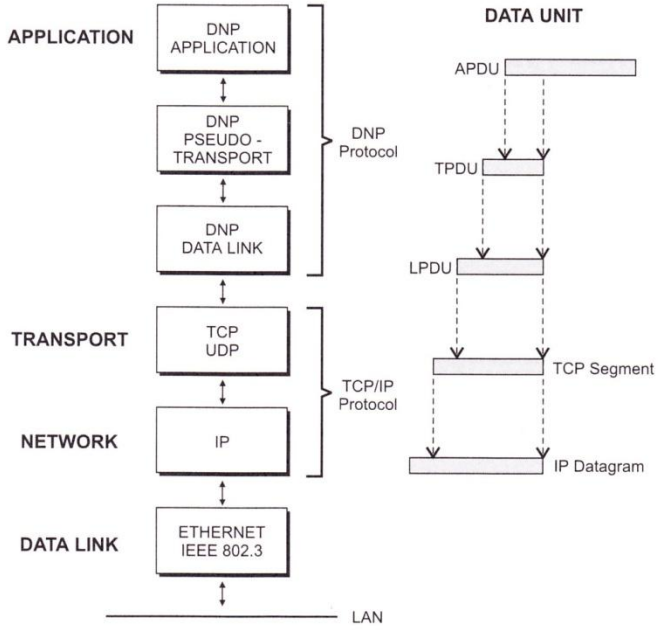

Figure 2-12: DNP3 over IP (reproduced from [44])

At the other end of the network, once the TCP/IP headers are stripped, the communication endpoint receives a fully formatted classical DNP3 packet, as if it had arrived from a connection over an RS-232 serial cable. This also means that, even if multiple hops are required to reach the communication endpoint, the various nodes in a SCADA network still act as if they are physically connected in a star pattern (in a single master design) or a tree (in a multiple master design). In that sense, it is important to keep the logical topology in mind when dealing with DNP3 networks.

Another issue that must be dealt with is the fact that, unlike serial communication, the typical TCP/IP architecture relies on a shared communication medium. When multiple nodes attempt to use the medium at the same time, some sort of mechanism must be used to manage the conflict. For Ethernet, collision detection with exponential back off is used. Exponential back off is based on a probabilistic model to determine the back off time. Concretely, this means that the transmission on Ethernet is not deterministic if there is contention on the communication medium. In order to avoid this, operators of SCADA networks attempt to deploy them in order to prevent contention (e.g. using switches or point to point communications, providing high bandwidth or quality of service, etc.). However, this property must still be kept in mind, especially if operating in quiescent mode. It is very possible to trigger multiple nodes to initiate

communication. For example, if one could trigger a failure that affects multiple nodes concurrently, the nodes would all attempt to send “status change” messages at the same time, possibly causing contention on the network.

2.4 Experimental approaches in SCADA experimentation

Even if new techniques are found to defend SCADA networks, it is imperative that they be tested to see if they are effective. Obviously, it is preferable to avoid doing those tests in the field, where the new security devices and techniques might interfere with operations. To solve this problem, a number of approaches have been proposed to provide an experimental framework for SCADA security research. This section, adapted from work we presented at the First International Symposium for ICS & SCADA Cyber Security [37], provides an overview of current approaches used for SCADA and ICS security research and of the limitations of those approaches to perform experiments focusing on network security.

2.4.1 Full Physical Deployment

One of the more realistic approaches to do research on SCADA and ICS systems is to actually deploy a system and perform experiments on that system. The National SCADA Test Bed [45], with its seven substations and 61 miles of high voltage transmission lines, is an example of this kind of implementation. This approach allows researchers to create experiments that have a high resemblance to real world systems, because it is using a full implementation of both the physical component and the software component. However, this approach suffers from a number of drawbacks for security research.

The first drawback is that deploying a real system requires significant investment, both in terms of capital and in terms of manpower. In terms of capital investment, let us consider Hydro Quebec's annual report [46]. The cost of replacement of the software for the management and analysis of the transport network is budgeted at 32 million Canadian dollars. This does not include any physical components (such as power lines and substations). This would suggest that the cost of standing up an at-scale laboratory is likely to cost tens of millions of dollars. In addition, SCADA equipment usually needs to be manually configured, requiring specialized knowledge to configure. This increases the manpower cost to stand up this kind of laboratory.

The bulkiness of physical equipment also creates a substantive lab space cost, especially in an academic setting.

A second problem is that of "decontamination" and reconfiguration of the experimental setup. Because real equipment is used, any modification to the original configuration, e.g. as a result of performing an attack on one of the machines, needs to be undone manually. This increases the costs of operating the test bed, increases the downtime of the test bed and may create unpredictable states if the decontamination is not thorough. This causes significant drawbacks for the repeatability of experiments.

2.4.2 Partial implementation

A possible compromise to reduce lab space use is to limit the scope of both the network studied and the physical equipment required. The SANS Institute, with their Cyber City project [47], followed this path. The computer network of a small town was reproduced on virtual machines, including the user profiles and actions, in order to be able to train experts in attacking and defending networks. This training includes SCADA systems that might be operated in a small city, i.e. the water treatment and transport systems. The SCADA components are connected to a small scale model town in order for the students to be able to observe the physical consequences of cyber attacks. For example, an attacker might send a false command and switch a railroad track, causing two model trains to crash together.

By limiting the scale to a small town, it is possible to create an environment interesting enough for students, while keeping it manageable, both in terms of manpower and real estate. The use of virtual machines allows for fast resets to initial configurations making decontamination straightforward. The physical consequences of attacks on ICS networks can also be very plainly observed. This provides a good environment for education. Unfortunately, the Cyber City model is limited in terms of possible research. As with physical test beds, Cyber City requires physical components, making it harder to do testing on configurations other than the default configuration. Also, addressing the problem of scale by scoping it to a small city prohibits any research done on problems with a larger scale.

Other implementations of small scale SCADA networks are common in the literature. Examples such as Dondossola [48], [49], Quieroz [50], Morris [51] and Hahn [52] provide a framework for

the use of small implementations for security research. As with Cyber City, these partial implementations suffer from a lack of scalability because they are limited by the equipment that is physically available. In addition, they suffer from what Hahn calls "configuration management" problems. In other words, it may be difficult to reset these systems to pre-experimentation configurations or to modify the configurations to alter the topology.

2.4.3 Software only

Another option for ICS security research is to use demo versions of SCADA software. A number of ICS vendors offer trial versions of their industrial control software. This software is usually a HMI application (software designed to allow human operators to interact remotely with industrial control equipment) with some missing functionalities such as a limited number of days the software can be used or a limited number of machines the software can interact with. This allows a researcher to observe communications that are properly formatted with minimal effort. As such, it is often used for research focused on protocol security (ex. [53]).

The major drawback of this approach is the lack of physical effects. While it is possible to hook a trial version to a couple of actual machines and turns lights on and off, it is not practical to use this setup to measure realistic physical effects. Unless great care is put into designing the physical network connected to the SCADA system, it is unlikely that the physical network will provide a realistic feedback to the SCADA system. For example, in a real system, turning off a breaker will shut down the power to the line making the sensor register a drop in voltage and possibly increase the load on power generators. In that sense, a network packet may very well have a scope of influence far greater than is possible to model with trial versions of HMI software.

2.4.4 Simulation

To solve the problems of scale with physical implementations, it is possible to use simulation. A simulation approach uses a model that is an approximation of reality to approximate the results of whatever inputs a user provides the system. The production of a valid simulation testbed for SCADA research is an active field as shown by [54], [55] and [56]. However, it is unclear how most of these simulators truly approximate a real network. Sometimes, the difference is because of a specific research focus that does not require high fidelity of results. For example, the TRUST-SCADA testbed [54] is focused on system level security research at the IED/PLC level

and does not require complex network interactions. Others are limited because they do not have the capability to have detailed models or actual data and have to resort to assumptions. For example, the simulation framework from Quieroz et al. [56], which expands on their partial implementation efforts, addresses the problem of scale by simulating components that cannot be integrated. But this technique cannot expand on the security metrics they are observing because the coarse grained simulation at the network level does not allow them to look at individual packets.

Davis et al. [57] introduced a SCADA test bed that provides a user with an electrical power system HMI that is plugged into a computer network simulator and Bergman [58] presented its use for computer security research. So, when a user sends a command to turn off a breaker for example, this network simulator reproduces the network packet and its delivery to the destination. Once it reaches its destination, the simulation software generates a real packet with a virtual IP address and sends it to the PowerWorld electrical simulator to see what effects the command has on the power flow. Power World can send packets back through the simulated network and ultimately be displayed on the HMI. In that sense, the physical effects of cyber attacks on the power grid can be observed on the HMI from the results of the power flow calculations. Unfortunately, the approach suffers from some drawbacks.

The first drawback is in terms of the validity of the model and the soundness of the measurements. Because the simulation is not using real SCADA equipment or network components, but a mathematical model of the equipment, there may be a significant difference between results observed in a simulation and an actual real world deployment. It is possible to validate the simulation models for both the network side and the power flow side to make sure they behave in a way similar to real world networks. However, security research has a tendency to deal with extreme or edge cases for which a model, even if it has been validated under normal operating conditions may react differently than a real implementation.

Another inconvenience is that the configuration required and data produced are in formats that are not directly portable. For example, the RINSE network simulator used in Davis et al. [57] is focused on coarse traffic metrics. So, data on the packet level is not always available. In that sense, results are less portable than if a more conventional packet capture file format, such as PCAP, was used.

2.4.5 Emulation

If a physical implementation is too expensive and a simulation does not quite allow us to represent a real network with the fidelity we require, we may consider emulation, i.e. a system that duplicates exactly rather than approximates the behaviour of a real system. For security research, the DETER test bed [59] and the Emulab network test bed with large-scale virtualization [60] are two examples of medium to large scale network emulation environments that could be used for ICS security research and training. In both of these cases, an environment very similar to a real deployment can be programmatically deployed in the test bed. Malware and attacks can then be tested without impacting real systems. If dedicated virtual machines can be used, an approach similar to the isolated virtual clusters from École Polytechnique de Montréal's SecSI lab [61] can also be successful.

Past experience has shown that an emulation approach can address a number of problems such as: containment of experiments, isolation from concurrent experiment interference, confidentiality and integrity of configuration and results, and the prevention of misuse of the test bed. Also, because deployment and experiments are run programmatically, it is easy to perform decontamination and reconfiguration efficiently. In the isolated virtual cluster architecture, decontamination is even more straightforward using VMWare snapshots. However, these approaches have a major drawback – the modelling of physical effects. Because all three test beds described above were designed to emulate cyber attacks, they only emulate digital electronic components. In that sense, it is even harder to model physical effects than with the use of trial software. Usually, in the operating environment of emulation clusters, it is physically impossible to install the custom I/O cards that can create the analog signals required by many PLCs or ICS machines.

2.4.6 Impact assessment

Another aspect of SCADA security research is the evaluation of the impact of security incidents. Operators of SCADA networks are reluctant to part with the details of any incidents they suffer (with rare exceptions such as the Maroochy water facility [4]) and the prospect of infecting a live system to test the effect of an attack is remote at best. In that sense, researchers with access to a

reasonable experimental test bed for SCADA systems have attempted to provide estimates of the impact of cyber attacks on SCADA systems.

Fovino and al. [62] have attempted to study the impact of malware designed to cause impacts in SCADA systems. To do so, they modeled a power plant based on observations made at a real-world site and created a test bed. Unfortunately, they did not model their attacks with the same level of fidilety, opting to emulate the attacks with a mobile agent simulator that replicates the behaviour of malware. While this level of fidelity may be adequate to draw some conclusions about network data, the observed metrics focus on system level or physical impacts (with the exception of the minimal DoS case study) with no evaluation of the impact of the implementation of their malware model or the middleware required to run their mobile agent. For example, they noticed that none of the worms they attempted to reproduce caused a system failure. It is impossible to tell if this is the result of resiliency in the SCADA network or the result of their malware simulator not interacting with the systems in the same way real malware would, perhaps locking up a thread, consuming all of the memory, modifying network paths and so on.

Another attempt by Sridhar [63] was made to assess impact of integrity attacks on SCADA systems. This study creates an analytical model of the attacks and integrates those attacks in a generic power flow balancing methodology. The assumption is that an operator would follow the methodology, come to an erroneous conclusion about the state of the system and perform an action that is contrary to his interests. A simulation is then constructed based on the analytical model to show that the analytical model performs as expected. Unfortunately, there is no validation of the model and the model seems to present serious limitations at first glance. Notably, there is no feedback loop that creates an electrical network effect based on the reaction of the operator. For example, if an operator is tricked into activating a breaker, the power flow will be diverted on other lines and this will trigger new measurements that are based on the ground truth and not on the falsified report. This would require the attackers to recompute the expected values for their falsification software faster than the actual system converges. At the same time, the power of the attacker is underestimated. In their model and attacker can only set a sensor to the minimum or maximum value of the sensor for a limited amount of time where in reality an attacker can send arbitrary values (even impossible ones) for an unlimited amount of time if he obtains administrative access to a machine.

The team of Bobbio and al. [64] have also attempted to study the impact of DoS attacks on the SCADA network. Unlike Sridhar [63], the attack model is very detailed and is based on an actual failure scenario. The network model is also very detailed because it is based on the actual systems. However, the study focuses only on the network effects of the DoS attack on the telco network. While the delay between the different SCADA nodes is calculated, we have no indication of how this delay affects SCADA traffic, if any critical packets are dropped or if operations are impaired in any way. In that sense, while the data might be useful for someone with a SCADA test bed to calibrate the network traffic generated by an attack, it provides little insight on its own.

Instead of building a test bed where simulated attacks can be reproduced, the SCADA honeypot by the SCADA HoneyNet Project [65], a subproject of the HoneyNet project [66], strives to observe SCADA attacks in the wild. The main advantage of a honeypot is that it is a trap designed to lure in actual attackers and observe how real attackers behave on what they think is a real system. The use of honeypots, which are not connected to live systems, can therefore allow defenders to gather valuable intelligence on attackers without putting live systems at risk. The downside is that, because there is no real system behind the honeypot, it may be possible for the attacker to be able to determine that he is facing a honeypot and adjust his behaviour accordingly. Another limitation emanating from the lack of connectivity to a real system is that it is not very useful to gather any real knowledge about the interaction between cyber and physical components. As a testament to this, the SCADA honeypot project has not produced any public reports of SCADA specific attacks even if attacks by groups such as APT1, a.k.a Comment Crew, have been reported on SCADA honeypots [67].

In summary, while each of these approaches have merits, they are not well adapted for research in SCADA network security. For some, the financial cost may be too high, for others the experiment setup time may be too long, the system might not be scalable, the cyber-physical interaction may not be correctly represented or the network traffic may not be of sufficient fidelity. This underlines the lack of an experimental platform for the realization of repeatable, high-fidelity network security research.

2.5 SCADA security research

Even with the lack of a platform for network security research, there are a number of researchers working to secure SCADA networks. These efforts come from a variety of fields, including information security, risk assessment, control engineering and economics. This section presents an overview of research efforts to address the problem of securing SCADA networks. We start by presenting research on offensive techniques, then we follow by research focusing on risk assessment and compliance and finally, we cover research on defensive techniques.

2.5.1 Offensive research

To fully understand the threats we need to defend against, a number of researchers have focused on researching offensive techniques to attack SCADA networks, or to map the vulnerabilities that are affecting SCADA systems.

The first line of research is the security analysis of the SCADA protocols. The research done by Dutertre [68], Edmonds and al. [69] and more recently by Hagen and al. [70] are good examples of research in that domain. In both instances, formal modeling was applied to a SCADA protocol (ModBus and ModBus over TCP respectively) to find weaknesses in the protocol that could be exploited by attackers. Similar efforts have been made to evaluate the security of other protocols such as DNP3. The taxonomy of DNDP3 attacks by East and al. [71] or of cyber attacks on SCADA systems by Zhu and al. [72] provides a very good overview of the efforts that have been made in that field. However, as seen previously, most attackers prefer to rely on the plethora of "traditional" vulnerabilities in SCADA networks which require much less effort to target. Because the formal modeling is used to help secure the protocols as well as for designing new attacks, it is not likely that protocol attacks will become a low-hanging fruit in the near future. Yet, these attacks are part of the toolkit of a skilled attacker and should be considered a viable option. As a matter of fact, research on defensive measures to prevent these attack often requires the development of attacks to have test cases (see examples Wang [73] and Gao and al. [74]). In such cases, it may be hard to determine if these attacks will indeed be representative of what malicious actors may come up with in the future.

In addition to the analysis of the SCADA protocols, there is also some research into SCADA application programs to evaluate their vulnerability. This work is done both by academics such as

Belletini [75] and industry researchers such as Internet Security Systems (ISS) [2] and TippingPoint [76]. It will come as no surprise that SCADA programs were not developed with security in mind. In the limited sample reviewed in their research, Maynor and Graham from ISS [2] observed numerous insecure coding practices such as the use of functions known to be vulnerable (e.g. strcpy(), sprint(), etc.), the lack of validation for untrusted input such as network traffic, widespread use of clear text and little or no ability to perform authentication. These results can easily be explained by the fact that security was never a design specification for code that was intended to be run only on segregated, trusted systems. Some security experts (such as Henry in [12]) even caution penetration testers to avoid performing network scanning on SCADA systems that are over 5 years old because of the high risk that such systems would crash if faced with a malformed packet. This indicates that the domain of vulnerability research for SCADA programs is actually a widely untapped field in public domain literature. It is the assumption of this author that the main limiting factors to the progress of that field of research are the lack of availability of SCADA equipment for such research and the fear of prosecution under harsh anti-terrorist laws for the researchers or the fear of providing adversaries with attacks. None of these limitations is likely to hinder a highly skilled and motivated attacker.

Lastly, there has been some research in the field of SCADA specific malware, which means malware that resides on SCADA specific hardware. The most common example is the development or analysis of so-called “smart grid worms” (such as [77], [78], [79]). These worms would use the new processing and storage capacity of smart meters to cause all kinds of havoc with the electrical distribution system. A lot of SCADA equipment tends to be built on top of generic off-the-shelf operating systems (such as the Windows OS) rather than on custom-designed hardware and does not even require custom malware. Industry reviews such as the one presented by ISS [2] testify to this by relating famous cases where SCADA equipment was infected by run of the mill worms such as Blaster and Sasser which targeted Windows machines.

While this kind is research allows us to better understand the problem space, it is clear that it does not provide much in terms of solutions for securing the electric grid's SCADA system. More importantly, most of the research is aimed at finding specific vulnerabilities. This provides little guidance in terms of creating an attack model against which to test our defenses. We must instead turn on the few technical descriptions of cyber attacks such as Maroochy, Stuxnet and Shamoon to build attack scenarios. Does it really matter if Stuxnet was using a ModBus specific

vulnerability instead of Windows-based exploits? And, while an advanced attacker might make use of the types of attacks developed in this research, it is still unclear what role these attacks would play in their penetration plan. As such, it is difficult to find direct application of this research in our problem space.

2.5.2 Preventing attacks in SCADA systems

Because SCADA networks have a number of idiosyncrasies, a great deal of effort is spent in the development of SCADA specific protection. Notably, research in SCADA cryptography, SCADA firewalls and SCADA-aware IDS require special attention.

Typically, SCADA protocols have not used cryptography because of the perception that limited computational resources and sensitivity to delay would be challenges too great to overcome. This did not cause a problem when SCADA systems were still isolated systems. However, when experts started to look more closely at the security of SCADA networks, many have identified lack of cryptography as a sign of lax security. Coupled with the fact that traditional cryptographic methods are regarded as being inadequate for SCADA, it is not unusual to see a good deal of effort invested to incorporate cryptography in SCADA networks. This work can take the form of developing new protocols or extensions into existing SCADA protocols in order to provide additional cryptographic functionalities (see examples [80], [81], [82], [83], [84], [85]). Another research axis has been to propose SCADA specific key distribution mechanisms (see examples [86], [87], [88], [89]). Even in light of this research, it is still unclear what the value of cryptography would be for SCADA security. In most SCADA deployments, confidentiality takes a back seat to availability and integrity. In multiple integrity attack scenarios, such as the physical compromise of a machine or the remote exploitation of a vulnerability in an application, adding encryption to communications provides no help. The data is in the clear on the physical device or the socket is open an accepts all packets. At the same time, cryptography may hinder other defensive measures based on packet captures such as network-based IDSes.

Another avenue for defensive research is the development of firewalls designed for SCADA systems. The idea is to strongly enforce perimeter separation between the SCADA production networks and the office networks. This is typically done by creating an application layer firewall that can parse SCADA packets and reject anything that isn't "expected". These firewalls are installed in front of all SCADA endpoints to intercept all the traffic going to SCADA equipment.

Additional functionality, for example the inclusion of a proxy that can perform authentication, can also be added. Examples such as patriotSCADA [90] and Tofino security appliance [91] are good examples of the state of the art in that matter. Bradetich [92] offers another solution to the problem by applying the architecture in use for secret level networks in the government to enforce the "air gap" with the SCADA network. More interestingly, Hadeli and al. [93], instead of attempting to build a complete protocol parser for SCADA, suggest using the fact that SCADA networks are more deterministic than normal office networks to build firewall and IDS rules. By looking in SCADA configuration files, they can build of list of expected communication pathways and generate rules based on this expected traffic. While a useful part of the defensive architecture, the current state of vulnerability of the SCADA networks and the fact that adversaries have multiple ways to bypass perimeters and jump air gaps makes the use of firewall insufficient to solve the security problems of SCADA networks. The numerous examples of infections from a USB key are a testament to this fact.

Instead of trying to integrate new security mechanisms to legacy systems, some research has been done in optimizing existing countermeasures to obtain better security for SCADA networks. In their work, Anwa and al. [94] have made significant efforts to optimize security for SCADA networks using a combinatorial approach of known countermeasures. That problem is shown to be a NP hard problem (can be reduced to the Multiple-Choice 0-1 Knapsack problem). To solve the problem, they build an analytical model of their attacker and of their network defences. They then choose a set of defences and apply them to their target network (in their cases a substation network). The set of defences that can be bought with a given budget represents their design space. Then, they calculate the worst case damage their model attacker could do on the network protected by the chosen defences. Using heuristics, they change the set of defences to cover the design space. The solution in the design space which allows the attacker to do the least damage is the optimal (or sub-optimal) solution. This approach is interesting because it takes into account the problem as a whole and uses a metric that is related to the controlled system (power delivery) instead of an information security or network metric (such as bandwidth available).

However, the paper has some limitations. The main problem is the limitation imposed by the complexity of the analytical model. To cope with the complexity, the authors limited themselves to only three possible defences (segregation by firewall, segregation by VLAN and link

encryption) and their attacker model is inexplicably crippled (limited to only one tap, limited to one type of attack and has no economy of scale). This produces results of limited applicability in the real world, especially since it is unclear how the model was validated.

Ultimately, all these efforts strive to reduce the vulnerability of SCADA systems. While this provides significant security gains against indiscriminate attackers, it provides less protection from attackers that systematically probe systems for vulnerabilities, such as advanced persistent threats, from eventually finding a way into the system. As such, none of these methods are sufficient to secure SCADA networks from advanced persistent threats.

2.5.3 Detecting attacks in SCADA systems

Instead of trying to prevent attacks, some researchers have focused on detecting attacks within a SCADA network. Peterson [95] presents a list of problems SCADA-based IDSes could address. For example proposing that SCADA vendors create rules to whitelist packets based on protocol adherence or that researchers dig in the large volume of historic data collected from SCADA meters to find anomalies. However, he does not provide any suggestions on how these might be implemented. He underlines the lack of support from traditional IDS sources for detecting attacks on SCADA networks. A number of researchers have attempted to provide a solution by building IDSes dedicated for SCADA environments.

One possibility is to attempt to detect attacks at the host level. In his research, Yang and al. [96] have attempted to build a host-based IDS that would detect a number of attacks based on computer performance metrics. Unfortunately, as they point out themselves, the data they use to test their model is inadequate. Oman and Phillips [97] instead focus on using the configuration files to see if alterations have been made, or if attempts to use a functionality that was not configured were made, to create a form of host-based IDS and configuration management system. Under these circumstances, an attacker sending malicious SCADA commands or exploiting a software vulnerability would not be detected, nor would malware that did not modify the standard configuration files. In that sense, the usefulness as an IDS focuses on a corner case of the attacker's reconnaissance and might only provide marginal usefulness.

A number of propositions have been made to detect attacks in the SCADA network by the detection of anomalies in the state of the controlled system. For example, Bigham and al. [98]

suggest using n-grams on the state estimation data to determine if the system is suffering from an anomaly. Carcano and al. [99] suggest using the "proximity" between two different network states to detect anomaly. A disadvantage of this type of technique is that it relies on the state estimation of the network which is obtained through the sensors installed on SCADA devices (see section 2.2). A subtle attacker is likely to alter those values, as was the case with the Stuxnet virus, and a blatant attacker is likely to cause disruption that would be identifiable as an anomaly even without the use of a sensor.

A number of attempts have instead focused on detecting attacks at the network level. Cheung and al. [100] and Goldenberg and Wool [101] focus on analyzing the Modbus protocol to create a model of the protocol and detect any deviation from it. In other words, forming a kind of white-list of acceptable Modbus states and transitions. Unfortunately, the SCADA worm has shown that malware that attempts to send false SCADA traffic is likely to respect protocol formatting. This limits the applicability of this type of intrusion detection to detecting protocol exploits. Instead of using protocol modeling, Schuster and al. [102] propose using machine learning techniques to learn patterns over time and detect any sudden changes in those patterns. While their paper presents the details of how learning would be implemented and what challenges are envisioned, including the problem of feature selection, there is no evaluation of the effectiveness of the solution because they have no datasets to test it on. Rusu and al. [103] attempt to solve the problem by generating a dataset with a network simulator. However, the values they are using for the amount of traffic generated, the type of traffic, and some of the proposed SCADA topologies are completely arbitrary making the validity of their results suspect. While the problem of the lack of good datasets for testing is prevalent in IDS research [104], the problem is greater in SCADA security research because of the sensitivity of production network data and the high cost of creating test beds (see section 2.4 for an in-depth discussion).

Some researchers have access to production network datasets and have attempted to propose IDS for SCADA networks. Hadeli et al. [93] use a parser of the configuration files for the SCADA protocol IEC 6185 to automate the creation of firewall rules to only allow legitimate traffic. However, their implementation is limited to creating filters based on IP addresses and creating IDS rules detecting missing traffic, for example the lack of traffic from a node that has been shutdown. Rather than using configuration files to leverage the determinism of SCADA network, Langill [105] parses packet captures. The goal is to create a map of legitimate communication

paths on a deployed system to create SNORT rules that detect communications falling outside of the legitimate pattern. Langill starts by merging the packet captures from all parts of the network. Then, he maps the communication pairs to create a matrix of allowed communications. Finally, anything that is not allowed is considered illegitimate and a snort rule is generated to detect it. While this approach improves on Hadeli et al., we still rely on a single characteristic of SCADA traffic: the logical tree architecture emanating from the application layer protocol. For example, the fact that all communications are originated from the MTU or that the protocol relies heavily on polling. Barbosa and al. [106], [107] have used data from actual SCADA deployments to produce communication frequency-based, and flow-based, anomaly detection and tests those detectors. Unfortunately, because they do not know the ground truth, they are only able to provide a performance evaluation of the number of alarms generated as related to each dataset. No evaluation of the number of false positive and negatives is provided. This indicates that even actual data from production environment might not be ideally suited for experimentation if the ground truth is not known.

In summary, research in detection of attacks in SCADA network suffers from a host of limitations. The most important of which is the lack of credible evaluation of performance. While the value of using the "determinism" of SCADA networks as a leverage to find attackers producing unusual patterns of communication has been suggested, approaches have focused on configuration information such as configuration files or common communication paths instead of using traditional network-based anomaly detection. We suspect this ties back to the lack of datasets which might be used to characterize SCADA traffic and evaluate its suitability for anomaly-based intrusion detection.

2.6 Conclusion

In this chapter, we have seen how the electrical grid can be considered as a collection of interconnected networks. At the center of the interconnection is the substation where power can be re-routed or distributed. These substations hold the majority of the control elements of the system. These control elements perform two basic functions: estimate the state of the electric grid and alter the state. SCADA networks are used to perform both these tasks with measurement points and control points respectively. This gives an attacker with control over SCADA equipment a great deal of power over the state of the electric grid. In addition, SCADA networks

provide attackers with a far greater scalability than physical attacks. This transforms sites where attackers could previously only do local physical damage, such as maintenance sheds and partner sites, into risks to the entire grid because access to the SCADA system is provided.

To address these risks, research has been undertaken, especially since the discovery of Stuxnet. However, a large section of the research effort has been focused on reducing the vulnerability of SCADA networks. Multiple approaches have been suggested such as improving firewalls or adding cryptography. While this research is useful in addressing threats to SCADA networks, it mostly serves in reducing the threat of indiscriminate actors that do not systematically search for vulnerabilities. Stubborn attackers, including advanced persistent threats, will persist until they find a way in.

When they do find a way in, we need to find them. A number of research thrusts in detecting attackers on SCADA networks have attempted to leverage the high level of "determinism" in SCADA networks to find attackers. Unfortunately, most of this research suffers from a lack of validation by providing unvalidated attack or traffic models or by abstaining from evaluating the performance of the suggested methods. For those that do not suffer from this lack, they have a limited reach because they must limit themselves to configuration files or commonly used communication paths. They cannot leverage traditional anomaly detection-based intrusion detection because they do not have access to a detailed model of SCADA network traffic. An anomaly-based intrusion detection system based on a detailed characterization of SCADA network traffic would provide a significant contribution to this field.

The problem of generating high fidelity data sets suitable for experimentation for cyber-physical system is hard. A number of researchers have tackled the problem, but none of the approaches have produced a method that can provide the network traces at the fidelity required to perform experimentation in network security. Each method presents major drawbacks in terms of either cost, repeatability of experiments, scalability, fidelity of the cyber-physical interaction, soundness of network data, or combinations of the above. As such, the elaboration of a method to generate experimental data suitable for the generation of high fidelity network data sets is a necessary contribution.

In order to create experiments that represent the real world reasonably well, we also need a model for the behaviour of attackers. Unfortunately, research in offensive security focuses on finding

vulnerabilities in SCADA networks and protocols. Very little is known about the tools, techniques and procedures attackers would use and no research focusing on this facet exists. As such the development of a model of an advanced persistent attacker that can be translated into attack specifications in an eventual network security experimentation is required prior to experimentation.

This study of prior art in SCADA security, in particular limitations pertaining to securing the network against advanced persistent threats, leads us in a trajectory toward using anomaly-based intrusion detection based on a detailed characterization of SCADA traffic. However, to reach this objective, we must create stepping stones in order to fill gaps in the current state of the art. We must first present a model of the techniques and procedures used by an advanced persistent attacker. Then, we must build an experimental environment that allows us to generate high fidelity network data sets. Finally, we will be able to use these data sets to create a characterization of SCADA traffic and test the performance of anomaly-based intrusion detection. All of this work will be presented in the following chapters.

CHAPTER 3 BEHAVIOUR OF ADVANCED PERSISTENT THREATS

Advanced persistent threats are often viewed as super hackers possessing quasi supernatural powers. This is even more true of hackers working for government intelligence agencies. This belief is often used as an excuse to avoid putting any efforts into stopping them. After all, these hackers are so good, nothing we mortals can do could possibly stop them. This aura is due in part because of the mysterious nature of the techniques and procedures used by advanced persistent threats. Prior to the detailed analysis of Stuxnet in 2010 and of the revelations about APT1, a.k.a. Comment Crew, in 2012, very little was known about the behaviour of advanced attackers. So, in order to stop them, we must create a model of advanced persistent attackers in order to be able to devise a defense.

Using parallels from previous asymmetrical conflicts, we proposed a model for the strategy nation states might pursue to engage in low intensity cyber conflicts. This model, which was presented at the NATO Cooperative Cyber Defense Center of Excellence Conference on Cyber Conflicts (now CyCon) in June 2010, insists that nation states actors will pursue a strategy for a high number of low impacts in order to prevent escalation. This puts the onus on stealth, enabled by the use of covert communications. Based on that realization, we build a defensive model focusing on increasing the capacity for surveillance which will constrict the attacker's actions if he wishes to remain stealthy.

This chapter presents an attack model for the behaviour of advanced persistent threats in SCADA networks and a model that serves as the basis for proposing surveillance as a defensive strategy. Section 3.1 resolves the apparent contradiction between the desire to target critical infrastructure and the lack of destructive incidents by evolving the concept of cyber warfare to cyber conflicts. Section 3.2 presents our contribution of a model for low intensity cyber conflicts, the pinprick attack strategy, that can be used to envision the behaviour of advanced persistent attackers against critical infrastructure targets. Section 3.3 presents the model of covert communication overlooked by a Warden as the basis for our strategy of increasing surveillance by providing more capabilities to the warden in order to limit the capacity of attackers.

3.1 Attacking the critical infrastructure

Modern attackers are often said to have abandoned the quest for fame to concentrate on the quest for profit. In that sense, it can be legitimate to ask who would attempt to disrupt critical infrastructure networks. This section, Section 3.1.1. looks at different actors and evaluates how they would target critical infrastructure. Section 3.1.2 details how the difficulty of assessing the risk of high impact and low probability scenarios affects this analysis. Section 3.1.3 reframes the nation state actor in the context of cyber conflicts rather than cyber warfare to address the incertitude in terms of risk.

3.1.1 Choosing critical infrastructure as a target

For a cyber attacker, targeting the critical infrastructure can have dire consequences. Since September 2001, in many countries, an attacker causing deliberate disruption of life support infrastructure is considered a terrorist. The willingness of law enforcement to pursue these attackers and the eventual penalties imposed on culprits are overwhelmingly greater than those of "typical" cyber crimes such as identity theft. In that sense, it is appropriate to examine the motivations that entice some attackers to choose to target critical infrastructure, in particular electric grids.

The first set of attacker targets critical infrastructure by accident. The distributors of mass-market malware and the users of indiscriminate exploitation tools typically invest little effort in targeting infrastructure because they are mostly interested in commodity resources, such as the bandwidth and processing power of compromised computers. For them, a computer in a critical system environment has no more value than the desktop PC in a cyber café. As such, it is in their best interest to avoid critical infrastructure systems which may trigger the wrath of the authorities. Unfortunately, the state of security controls in SCADA networks is such that the indiscriminate attacks sometimes get in. However, this class of attack is uninteresting in terms of engineering research because the countermeasures to effectively respond to this kind of attack are well known and the adversary is unlikely to aggressively pursue these targets if he is removed from the system. His efforts are better invested pursuing low-hanging-fruit systems that will require less investment and less risk for the same reward.

The second set of attackers deliberately targets critical infrastructure. Some examples of this type of attackers are extortionists, hacktivists and nation states. These actors are unfazed by the harsh penalties associated with the cyber terrorist label either because they imagine the reward worth the risk or because they feel they have moral obligations to their cause or their country. Unlike the indiscriminate attackers, if these adversaries are repulsed, they are likely to return instead of looking for easier prey. Except for the extortionist, increasing the cost of the attack by adding more defenses is unlikely to deter them because they are not after commodity resources. For example, it is hard to put a monetary value on freedom of speech and say that, past a certain threshold of investment, freedom of speech advocates will cease their activities.

Of the adversaries that cannot be deterred or redirected to easier targets, hacktivists are the least likely to cause dramatic impacts. After all, the main motivation behind hacktivists is often some sense of greater social good. As such, it is unlikely that they would resort to actions that would severely harm "innocent bystanders" because that would hurt their cause. For example, hacktivists motivated by environmentalist beliefs might be inclined to attack an "evil oil company", but they are unlikely to deliberately trigger an oil spill that would have serious consequences to the environment. In that light, it seems that the strategic aims of hacktivists would be suited better by avoiding the control systems where the risk of collateral damage is high and concentrate on attacking the corporate networks where confidential information is stored and where publicly visible targets, such as web servers, can be exploited for publicity.

This leaves the nation state sponsored attackers that will not be redirected or deterred and that may want to wreak physical havoc. Since the Napoleonic era, total war, which is the mobilization of entire nation states for conflict targeting not only the armed forces, but also the civilian infrastructure that sustains the armed forces, is considered a legitimate form of warfare. One such infrastructure is the power grid. Therefore, we can assume that nation states are interested in targeting the power grid for its strategic value. At the same time, the energy market is worth trillions of dollars [108] and is often the domain of national monopolies or large national champions. So, even in times of peace, some nation states might feel tempted to provide competitive assistance to national economic interests and may even employ underhanded tactics to do so.

3.1.2 Low probability/high impact scenario

When cyber warfare is invoked, images of catastrophic devastation immediately come to mind. TV shows predicting massive power outages in the middle of winter, nuclear meltdowns and massive flooding from dams are not considered possibilities, but certainties. Even if we disagree with those assessments, we must concede that sustained attacks on critical infrastructure can have a large impact on people depending on those infrastructures. At the same, it is clear that such attacks are not happening every day. The best documented example of a nation state sponsored cyber attack, Stuxnet [22], did not look like these Armageddon scenarios. On the other hand, with the level of control they achieved, it is likely that Stuxnet's handlers could have cause more damage, possibly even a radioactive spill. So, the possibility exists, even if the likelihood is low.

Risk can be defined as the expected loss of a given scenario. As such, we can calculate risk as the product of the probability of the scenario and the impact of the scenario. In the case of the cyber warfare scenario, we have an incalculably large impact and an incalculably small probability of occurrence. This situation is similar to terrorist threats for which risk analysis cannot fully guide policy makers [109]. We cannot evaluate $0x\infty$, so we cannot provide a numerical calculation of risk. In the same vein, it is impossible to quantify the amount of effort we should expend to defend against this risk. In that case, how do we guarantee the reliability of the power grid in this context? To address this question we must build a more reasonable model of what an attack on the critical infrastructure in the context of cyber warfare would look like.

3.1.3 From cyber warfare to cyber conflict

The vivid scenarios associated with cyber warfare depend on a number of strategic assumptions. One of those assumptions is that causing such damage would be in the strategic interest of an adversary. It falls within reason that a cataclysmic attack, cyber or not, on a nation's critical infrastructure would be considered an act of war. The consequences of such an act for the aggressor would ultimately be unpleasant if the victim, or its allies, have any kind of retaliatory capacity. At the very least, it would invite conventional war from the victim. For a nation state engaging in *realpolitik*, the benefits gained from waging such an assault should outweigh the eventual consequences. Unless the victim and the attacker are locked in a state of total war, it seems unlikely that the balance of advantages and repercussions will favor such behaviour.

The existence of Stuxnet testifies to the willingness of nation states to sponsor cyber attacks even when not locked in a state of total war. In fact, the decades long Cold War, where no conventional military engagements between the two protagonists occurred, illustrates how nations can operate in an adversarial mode in a conventional setting without resorting to total war. Conventional military forces have developed doctrine designed to address the types of engagement they may face that are different than total warfare. The Canadian version, described in Canada's Army [110] and Land Operations [111], present an entire spectrum of possible military involvement, be they rescuing flood victims (military operations other than war) or waging war (warfighting). Figure 3-1 presents the full spectrum of warfare.

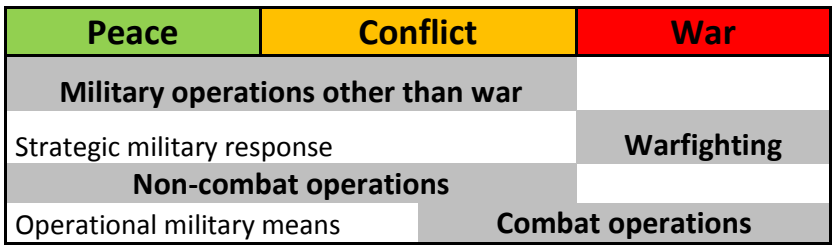

Figure 3-1: Spectrum of warfare

The type of military involvement is proportional to the degree of conflict in which the country is entangled. We could reasonably assume that a nation state would apply a similar approach to cyber warfare and engage in varying degrees of intensity depending on the degree of conflict. This brings us to evaluate scenarios less dramatic than the catastrophic cyber attacks, but more adapted to a world at relative peace.

3.2 Cyber conflict model

In the world we live in today, full scale warfare is uncommon. To build a credible attack model for a nation state actor, we have to build a model suitable for cyber conflicts. In this section, we present our contribution to the development of a cyber conflicts model: the pinprick attack. Section 3.2.1 presents the model for slow, gradual degradation as a valid offensive strategy. Section 3.2.2 test the model by analyzing how close the operation of Stuxnet was to the behaviour predicted by the model. Section 3.2.3 looks at current incidents to gauge how the situation evolved since the release of Stuxnet.

3.2.1 Pinprick attacks

This section is adapted from work published [36] at the NATO Cooperative Cyber Defense Center of Excellence Conference on Cyber Conflicts (now CyCon) in June 2010, three months before the publication of a comprehensive analysis of Stuxnet.

Pinprick attacks are an illustration of what can be done with low intensity cyber warfare. With Pinprick attacks, the trick is for the attacker to lead the defender into believing he is facing unconnected single instances of small attacks. This is done by staying under his correlation threshold. It is similar to the practice of "slow slicing" or "death by a thousand cuts" in the sense that you do not perform a single crippling attack, but instead a collection on non-crippling attacks whose effects add up to create the crippling effect.

In our pinprick attack scenario, individual damage per incident is low. It is therefore ill suited to attack hardened targets built with resilience in mind such as military communications. However, because it is a long-haul strategy, we can perform attacks on select points which will yield good results. The specific targeting of ball bearing factories by U.S. bombers in World War II is an example of operations designed to destroy a fighting capability without actually directly targeting military hardware. Can such an operation be carried out in a cyber warfare context? RAND's publication "Measuring National Power in the Postindustrial Age" [112] offers us some insight into how this could be done. This report presents a methodology to evaluate a nation's power using more than military power as the sole criterion. In the RAND model, combat proficiency is a result of the combination of strategic resources and the capability to convert these resources into military power. The easiest example is the case of military technology. A country with rich resources in terms of knowledge and money (strategic resources) can transform these resources into military technology through its military-industrial complex (conversion capability). Because we are talking about a combination, affecting either the resources or the conversion capability will result in a decrease in military power. We could present our "death by a thousand cuts" scenario as gradually injecting grains of sand into a complex clockwork mechanism in order to make it stop, or at the very least run less efficiently.

Defence from this scenario, in western countries, is mostly under the control of the private sector. For example, privately owned banks control most of the financial system, privately owned power companies supply the power, privately owned companies produce most of the technology and

hardware used by the military. The goal of these companies is to make profits. This objective is usually incompatible with spending money to defend against an unlikely scenario (e.g. cyber warfare). Increased spending for cyber security can even be detrimental to the health of a company. After all, if your costs are higher than those of your competition because of high security measures, customers will buy your competitor's products. This breeds a vulnerability rich environment which drives the costs of creating an attack operation down even in the face of government mandated vulnerability reduction programs. Attackers have all the time they need to perform exhaustive searches for vulnerabilities because the attack follows a deliberately slow tempo. This gives a determined attacker the agility required to attack only targets of opportunity and to follow the path of least resistance and pick the low-hanging-fruit. In that sense, a vulnerability reduction program does not offer adequate protection against pinprick attacks.

An important aspect of pinprick attacks is to keep the defender unaware that the attacks he is seeing are part of a coordinated strategy. As long as he is not able to correlate the attacks, there is no theoretical limit to the amount of damage you can inflict. This can be explained by the fact that, compared with each incident in isolation, the cost of coordinated response will always be higher than the incident's damage. For example, if you find a Trojan horse on a military contractor's computer, you clean it and try to assess the damage. If you find one on someone else's computer next week, you will do the same. However, if you find a Trojan on the computers of all the military contractors, you might take more active measures to stop whatever is going on. So, by design, pinprick attacks are difficult to defend against by centralized data correlation agencies such as CERTs.

Because pinprick attacks reside in the low intensity part of the spectrum, they are not well suited for what we consider warfare scenarios which require speedy conflict resolution. However, it is ideally suited for competition between near peers where one of the peers wants to slow down the progress of his other peers to catch up with them or increase its advantage.

Let us consider the fictional scenario where the countries of Alpha and Beta are near peers. However, the people of Alpha possess a significant advantage in technology over Beta. This advantage in technology allows the military of Alpha to hold a strategic advantage over Beta's military force, even if both are similar in other aspects. If Beta were to pursue a high intensity cyber warfare strategy, Alpha could respond by cutting its connectivity to Beta and escalating to

a military conflict where Alpha has the advantage. This course of events is therefore detrimental to Beta. However, Beta can instead decide to be patient and use pinprick attacks. Slowly but methodically launching attacks to undermine the confidentiality around Alpha's technology. Beta can sum the benefits of all his attacks (plans captured by a Trojan Horse, information recovered from a stolen USB key, communications intercepted on the wire, etc.) to catch up with Alpha in technology and negate Alpha's strategic advantage. It is unlikely that Alpha would recognize that the various incidents are connected to a coordinated effort by Beta to negate a military advantage because individual incidents only cause limited damage.

3.2.2 The case of Stuxnet

The pinprick attack model predicts than attacks from nation states will take a slow approach to avoid detection and continue doing small amounts of damage over a long period of time. The emphasis of the operation would be on not getting identified as a coordinated attack rather than on the destructiveness of the attack. The damage would be focused on disrupting military means at the source by restraining the supply of critical resources rather than directly attacking the end product. Finally, the attack would take advantages of multiple attack paths, picking all of the low hanging fruits in turn.

The political context in which Stuxnet occurred is a context of conflict between Iran and the majority of western countries over the alleged pursuit of nuclear capabilities by Iran. The conflict was escalating with some countries, notably Israel, starting to think about military strikes in Iran. In terms of spectrum of conflict, the protagonists were in the second half of the conflict region. This would be the area where pinprick attacks would occur: sufficient conflict for hostile actions, but not enough to require high tempo operations in support of kinetic warfighting. Our expression of the pinprick attack model predated the discovery of Stuxnet. As such, we might consider our pinprick attack model to have made a prediction on the unfolding of attacks by a nation state. We can consider Stuxnet to be a real world experiment of our model. If Stuxnet follows the template for pinprick attacks, the prediction is accurate and this lends support to the validity of our model.

Stuxnet's damage was subtle in nature. By altering the spinning speed of the centrifuges, Stuxnet altered the composition of the finished product of the enrichment process and made it unsuitable to use for military purposes. This was done in a way that is harder to detect than if the equipment would just cease to function, which would immediately trigger the suspicion of Iranian engineers.

In fact, great care was taken to cover the damage done. The inclusion of a rootkit targeting embedded control software is the proof that significant effort was made to make it hard for engineers to diagnose the problem. Detailed analysis of the state machine by Fallieres et al. [22] even shows that the machine can wait for days before starting its sabotage process. This is a deliberately slow tempo and a deliberate focus on stealth. This conforms to the pinprick attack model.

The targeting of centrifuges also conforms to the pinprick attack model. The process of enrichment is a critical part of developing nuclear capabilities. Much like planes cannot be constructed without the requisite ball bearings, it is not possible to build a nuclear bomb without fissile material enriched to a high level. Therefore, crippling the enrichment process directly cripples the capacity to build an atomic bomb. In addition, Stuxnet caused the centrifuges to prematurely wear out. Since this type of equipment is not readily available to a country under international sanctions such as Iran, Stuxnet also attacked the supply of material to the enrichment process.

One area where Stuxnet did not conform to the pinprick attack model is on the systematic picking of low-hanging-fruit. The fact that the target was not directly accessible from remote location may explain this discrepancy from the model. After all, there are not a large number of vulnerabilities available as ingress points for the cyber weapon and, once Stuxnet was firmly established, there was not a lot of incentives to find other vulnerabilities. In terms of attacking other resources required for building a nuclear capability, financial assets for example, it is very hard to provide credible facts proving their existence or lack thereof. Unless caught red-handed as was the case for Stuxnet, it is unlikely that any attack would have been publicised by either the perpetrator or the victim. In that light, the fact that Stuxnet did not conform to the model does not necessarily deter from the validity of the pinprick attack model.

3.2.3 Raising the bar

Once Stuxnet became public, many speculated that it would invite a number of copycat attacks based on its now public code triggering a sort of cyber weapon proliferation. In fact, Stuxnet's influence may be even more far reaching.

There is little doubt that a spy caught in the act of actively sabotaging a nuclear plant would invite serious consequence to himself and his sponsor. Depending on circumstances, it may be even considered an act of war. In that light, is Stuxnet an act of war? Since Iran did not declare war on the United States, *de facto*, it was not an act of war in this particular case. However, we can wonder about what the answer would have been if Stuxnet's sponsor had not been a superpower or a country protected by one. In any case, the fact that Stuxnet was not considered an act of war establishes a significant precedent. It could be argued that anything, up to and including Stuxnet, would not be considered acts of war. In consequence, in many circumstances it may not be appropriate to respond with force to such attacks. This constrains deterrence of cyber attacks.

If force cannot be a suitable response, the obvious response would be a response in kind. This means another cyber attack. In fact, some argue that Shamoon is a retaliatory strike from Iran [29]. The technical description of the Shamoon malware [28] reveals that Shamoon is less focused on stealth and more focused on destructiveness. Following the spectrum model of conflict, this could be construed as an escalation of the conflict. Even with its destructiveness, Shamoon was not considered an act of aggression. This, again, raises the bar for what is considered acceptable behaviour. In fact, we may only know where the line is when that invisible line is eventually crossed. Until then, it is reasonable to assume that this kind of behaviour will continue. As such, we feel that protecting critical infrastructure in general, and the power grid in particular, from targeted attacks from state sponsored actors is relevant.

3.3 Defensive strategy

In the face of mounting tensions in cyber space, it is clear that advanced persistent attackers, including nation state sponsored attackers, have targeted the power grid and other infrastructure. If our goal is to defend systems against these threats, we must devise a defensive strategy to counter strategies focused on stealthy attacks pursued over a long period of time. This section presents a defensive strategy focusing on limiting the attacker's ability to communicate covertly. Section 3.3.1 provides an explanation for the preference of covert communication by the attackers. Section 3.3.2 presents a communication model that models covert communications and provides the basis for enhancing Warden capabilities as a defensive strategy.

3.3.1 The use of covert communication by advanced attackers

The linchpin of the pinprick model is preventing escalation. Once a protagonist is intent on warfighting, it becomes likely that a conflict would not be restricted to the cyber realm. In the face of the strength of militaries and of collective defense agreements, the threat of escalation makes a significant deterrent for the most egregious attacks. It is doubtful anyone would think that causing a nuclear meltdown in the United States would not engender a significant response. At the same time, as seen in section 3.1.1., critical infrastructure presents an attractive target.

As a matter of fact, a number of incidents have suggested that advanced persistent attackers have specifically targeted critical infrastructure. The Mandiant report [33] identifies the energy sector as a top target. Krebs [32] talks about the stealing of SCADA software code. Chinese hackers have been caught in decoy water plants [67]. We can also mention Stuxnet [22]. So, clearly, deterrence does not prevent sufficiently motivated attackers from targeting systems like aqueducts and nuclear power plants. A possible explanation for this behaviour is simply that they did not expect to get caught.

As discussed in section 3.2.2, Stuxnet had a definitive emphasis on stealth. In that sense, significant effort was expended to avoid getting caught. Most of these efforts were targeted at the engineering telemetry (the malware playing back legitimate sensor values). Some effort was also invested in disguising the communication going to the Internet with the use of a covert communication channel. Presumably, this was done to prevent defenders looking at the traffic going to the Internet from identifying that there was malware inside.

If attackers are investing in stealth everywhere defenders look, we can expect attackers to make increased use of covert communications if we enhance detection of malicious network traffic. So, to build our attacker model, we must have a model of covert communication.

3.3.2 Communication model

In order to counteract the defenders' actions, attackers wishing to maintain a persistent presence in a system require frequent communications with the systems they have compromised. They have to update their tools, exfiltrate data, examine telemetry to gauge the defenders actions and so on. The more communications the attacker can establish, the more power he has over compromised systems. On the other hand, the more communications he has, the easier it is for the

defender to notice something is amiss. In that sense, there is an explicit trade-off for the attacker between bandwidth and stealth.

In their paper, Smith et al. [113] provide a mathematical model for the stealth/bandwidth trade-off based on the probability of detection of the defender. The higher the proportion of injected symbols to natural symbols for a given message size, the higher the probability of detection. If an attacker is intent on maximizing his stealth, he can deliberately reduce the proportion of injected to natural symbols to reduce the detection rate to an arbitrary level. However, this significantly reduces his bandwidth, and therefore his ability to react to the defender's actions. So, forcing the attacker to squeeze his bandwidth may prove a viable defensive strategy against stealthy attackers.

The typical model to represent this situation is Alice and Bob, two prison inmates in different cells, attempting to communicate escape plans in the presence of a prison warden. Figure 3-2 illustrates this situation.

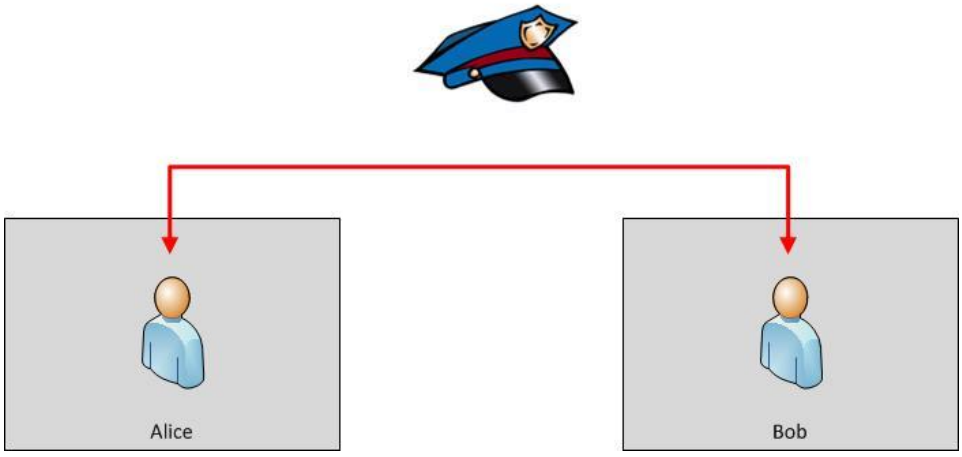

Figure 3-2: Communication model

In the case of network communication, we can model the communication between Alice and Bob using the Shannon representation of a channel as shown in Figure 3-3.

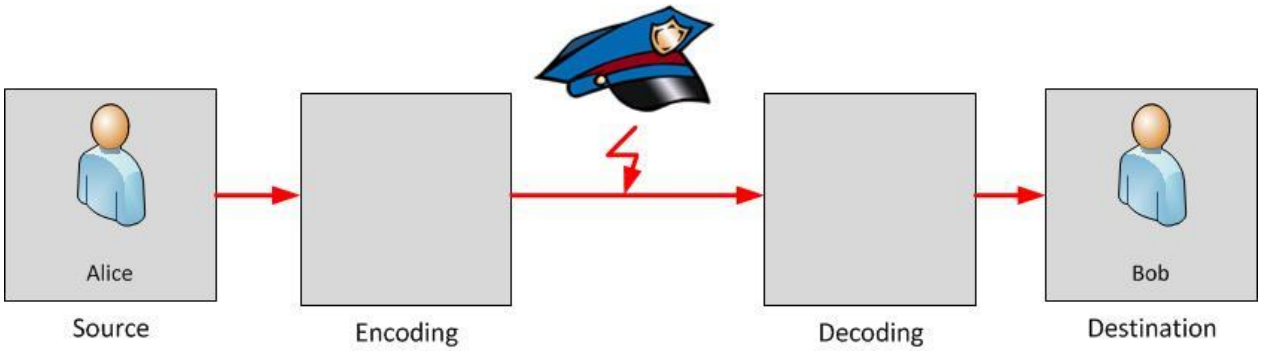

Figure 3-3: Shannon-based communication model

In this model, the warden intercepts network traffic between Alice and Bob and attempts to determine if the traffic is legitimate or malicious in nature. This is similar to the role and mode of operation of a network-based IDS. So, let us use a NIDS as a warden model.

If a malware signature-based NIDS, such as Snort, is used, any encoding of malicious traffic for which no signature rules exist will not be detected by the warden. In that sense, the attacker only has to find a previously unknown exploit to either carry traffic or to bypass the NIDS to be completely undetectable by the warden. The number of combinatorial possibilities to craft these exploits makes it unlikely that a signature-based NIDS rule set will have sufficient coverage to make this task difficult for the attacker. In that sense, a signature-based NIDS significantly limits the capabilities of the warden to detect new covert communication.

An anomaly-based NIDS warden builds a statistical model of legitimate traffic and analyzes conversations between Alice and Bob to see if the conversations follow the statistical model described. If it does, the conversation is judged to be legitimate and if it does not, the conversation is judged to be malicious. Following that rule, a malicious conversation must attempt to match legitimate conversations as closely as possible in order to remain undetected.

If the warden is an ideal warden, it will possess a complete description of the traffic. As such, traffic going through the channel will be required to strictly adhere to the description. This means that the traffic will be required to precisely follow any deterministic parts of the protocol (i.e. no exploits in the signaling of the protocol) and will be required to have the same entropy as legitimate communications. Using Shannon's theorem, the maximum quantity of information that should be carried on the channel is the entropy of the source. If the encoding adds entropy, for example with random padding, addition of timestamps or randomly generated sequence numbers,

we must also add this entropy to the entropy of the source. We obtain a channel entropy described by the following equation:

$$H(channel) = H(Source) + H(signaling)$$

According to this equation, an attacker that is operating in a network where the entropy of the sources is low, and the sources are using a protocol which introduces little entropy in the signaling, will have less bandwidth at his disposal than an attacker working in an environment where both of these are high.

Real NIDS are not ideal wardens. It is very rare that the traffic is sufficiently characterized to have the entire description of the legitimate traffic. Typically, an anomaly-based NIDS will focus on a small number of statistically significant traffic features that are highly indicative of anomaly. For example, consider a feature recording the values of two flags which cannot be set at the same time. In normal traffic, the proportion of traffic that demonstrates this characteristic will be zero. In this case, the decision to label this traffic malicious if this feature records that both flags are set is easy. On the other hand, if a feature can have a wide range of expected values, the feature is less relevant in terms of decision making for an IDS. In that sense, the easier it is to produce features that are relevant to decision making, the easier it is to approach the ideal warden.

Using this model of covert communication, we can focus on a strategy for fighting covert channels. In traditional study of covert channels, the complete elimination of side channels is eschewed in favored of limiting the amount of information that the attacker can transmit. By strengthening the capabilities of the warden, we can pursue the same strategy, forcing the attacker to trade further bandwidth for stealth. The more the attacker bandwidth is reduced, the more complicated it is for him to perform routine actions, such as updating his tools and increasing his presence in the system. So, while it will not prevent a persistent attacker from eventually establishing a presence in the system, it will restrict the impact of the penetration. Also, given sufficient constraints, it may well be that a mistake on the part of the attacker or a new operational requirement forcing the use of more bandwidth will allow the defender to ultimately detect the intrusion.

3.4 conclusion

In this chapter we presented a model for the behaviour of advanced persistent threats in SCADA networks. By showing that there is significant incentives for targeting critical infrastructure, we showed that some motivated attackers are likely to try their hand at attacking, for example, the power grid. The lack of spectacular cyber incidents involving critical infrastructure can partly be explained by the current state of international relations where there is no open state of warfare. This does not preclude the presence of low intensity cyber conflicts, under the threshold of full on cyber warfare.

In order to understand the impacts of this state of affairs on the strategy of advanced persistent attackers, we created a model for a strategy focusing on a high number of small impact attacks called pinprick attacks. This model was presented at the NATO Cooperative Cyber Defense Center of Excellence Conference on Cyber Conflicts (now CyCon) in June 2010 as a contribution to the community. Under this model, the greatest constraint on attackers is the motivation to avoid an escalation of the conflict in the physical realm. So attackers must focus on slow degradations requiring long presences in the adversary's network. Because of this, these attacks put a premium on stealth.

Based on the desire of the attackers for stealth, we presented our defensive strategy around denying that stealth. Using the propensity of attackers to express this stealth through covert communication, we offered the model for covert communication in the presence of a Warden as the intellectual basis of our defensive strategy. Then, we conclude that, by strengthening the Warden, we will be able to limit the bandwidth of attackers wishing to remain stealthy, constraining further their ability to perform routine actions such as tool maintenance and propagation through the network.

In order to test the effectiveness of that strategy, we will first need to provide a framework for the realization of network security experiments.

CHAPTER 4 IMPROVING THE FIDELITY OF SCADA NETWORK SECURITY EXPERIMENTAL METHODS

By analyzing advanced threat behaviour, we notice their propensity for covert communication and identified attacking their ability to do so as a valid defensive strategy to defend SCADA networks. However, because current research is not focused on this particular problem, it is not possible for us to use currently available research methodology to test the effectiveness of the strategy we propose. For the same reason a biologist requires appropriate foliage to evaluate the ability of a chameleon to blend in, we need a method to provide high fidelity network traffic in which attackers can hide. Only then will we be able to gauge our ability for finding covert communication in SCADA network.

There is a lack of good data for experimentation in SCADA network security. As shown in section 2.4, there is a lack of public domain data sets for SCADA networks and the current experimental methods are not adequate to provide high fidelity network traffic. Obtaining this data is a necessary step in devising an experiment to test our defensive strategy focusing on increasing surveillance in SCADA networks. So, we must devise a new methodology to generate high fidelity network traffic and implement an apparatus to generate the data. The validity of our approach must also be tested to ensure suitability for experimentation.

This chapter presents a novel approach combining emulation and simulation to generate high fidelity network data for experimentation. This work significantly advances the ability of the community to perform research in SCADA network security and sections of this work were presented at the First International Symposium for ICS & SCADA Cyber Security in 2013 [37]. Section 4.1 presents the ICS sandbox approach and its implementation. Section 4.2 presents training sessions in which the ICS sandbox was used, and which can be used as a benchmark for the fidelity of the emulation component in representing a real SCADA network. As further validation, section 4.3 reproduces a power engineering experiment using the ICS sandbox as a proof of concept of the hybrid emulation/simulation approach.

4.1 The ICS sandbox

This section is adapted from work presented at the First International Symposium for ICS & SCADA Cyber Security in 2013 [37].

Based on our study of existing approaches, we found that physical implementation-based approaches are too costly, but simulations cannot fully capture the interaction between the physical and computer system. Emulation approaches seem to provide the correct balance between realism and feasibility. However, they struggle to integrate the physical aspect. Our approach strives to find a way to integrate the ICS physical component with existing emulation infrastructure in order to create an ICS sandbox.

The goal of this ICS sandbox is to study the effects of network attacks, such as denial of service, falsification or injection of data, malware infection and so on, on both the network infrastructure of SCADA networks and on the power grid. In other words, the goal is not to find and test vulnerabilities in specific equipment, but rather to perform impact assessments of known attacks or to evaluate the effectiveness of network defences to detect or prevent these attacks. This distinguishes us from other works in emulation, such as Davis et al. [57], which focus on the behaviour of SCADA equipment and do not offer the granularity of network traffic necessary to perform network security research. In that sense, our approach is, as far as we know, the only methodology available for high risk network security experiments for SCADA systems that takes into account the physical side of the problem space.

4.1.1 Scoping

The first step is to scope the project in order to elicit requirements. The focus of our ICS sandbox is on network security. In that sense, only the elements relevant to network security are required to be fully emulated. Our focus was to make sure the network traffic that can be observed resembles as closely as possible that of a real-world implementation. In addition, any system component directly interfacing with the network, i.e. clients and servers, needs to be as close as possible to real-world implementations. The requirements of any other elements in terms of fidelity are less severe.

In terms of SCADA systems, we require the actual network to have the highest degree of fidelity, MTU and RTU machines to have a good level of fidelity and the HMI, PLCs and the actual physical system require less fidelity. In fact, for all intents and purposes, the physical system can be considered a black box where the inputs are values of control points (ON/OFF values for breakers and voltage or current values for set point controls). To achieve this, we chose an architecture such that in the core, where fidelity requirements are high, an emulation approach

similar to DETER [59] is used and in the edge, where less fidelity is required, a simulation approach similar to PowerWorld [57] is employed.

4.1.2 Implementation

For the core of the network, we require a suitable platform for emulation where we can run actual SCADA software and perform real-world attacks. We decided to adapt the test bed for high risk security experimentation and training proposed by Calvet et al. [61].

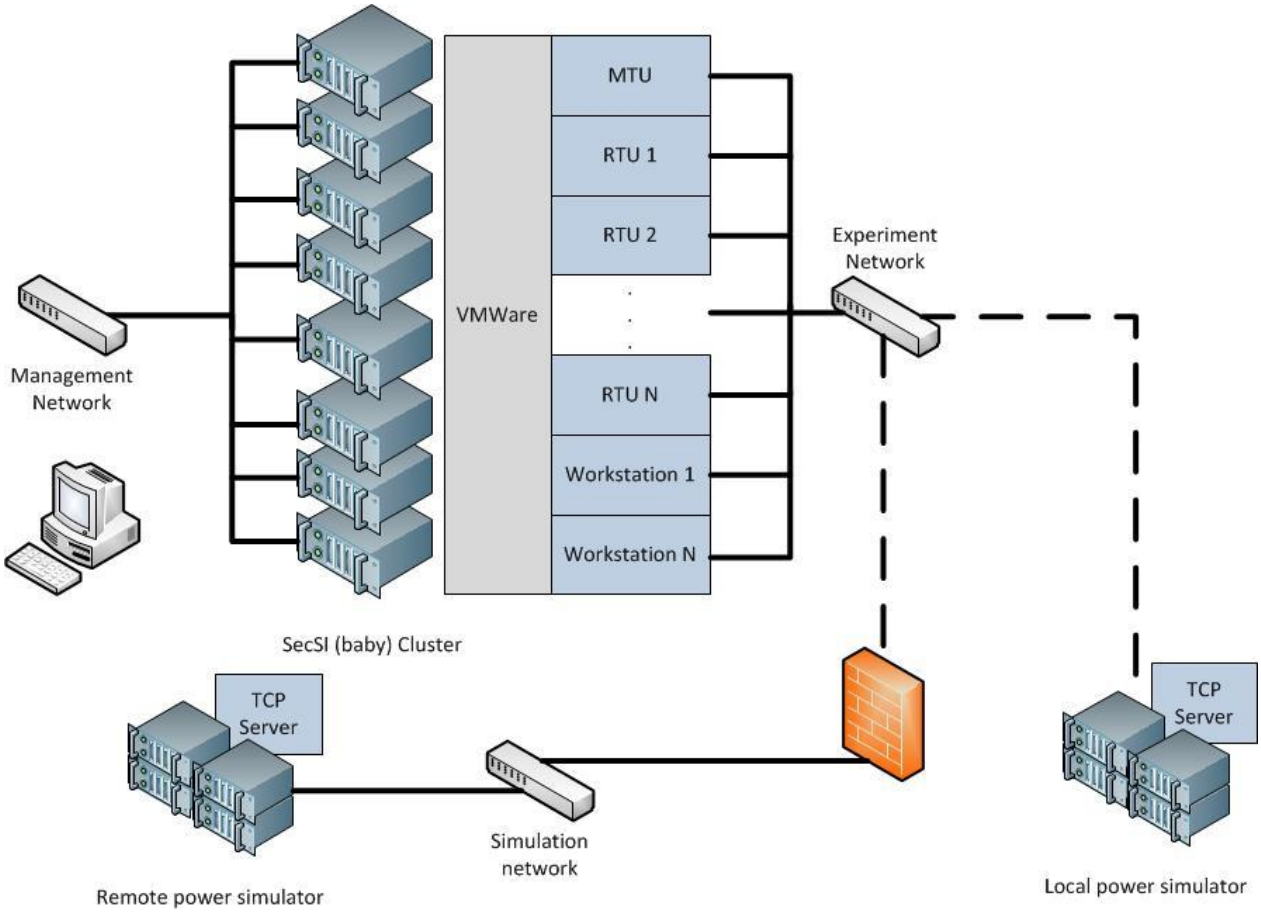

Figure 4-1: ICS Sandbox architecture

Our infrastructure employs a number of IBM Blade servers running VMware software for virtualization. A management network allows the deployment of experimental configurations (deployment of machines, starting/stopping the VMs, setting IP addresses, etc.) through the xCAT scripting language as described in Calvet et al. [61]. Figure 4.1 shows the architecture of the ICS sandbox. The entire SCADA system is emulated on virtual machines running on the SecSI cluster. The SCADA system is then connected through TCP requests to the electrical

power flow simulator. Each section of the infrastructure will be covered in detail in the following paragraphs.

The experiment network itself is a physical Ethernet network. Other network technologies could eventually be used to experiment with other types of interconnection technologies. Managed switches are currently used because the network topologies are simple, but virtual switches that could be configured programmatically by xCAT could be used. Port mirroring is used to capture and observe the network traffic so great care has to be applied in making sure experimental traffic makes it onto the network devices. Concretely, this means that communication between multiple virtual machines on the same server should not be implemented using the virtual networking provided by VMware unless this was specifically designed in the experiment plan.

Running on this physical infrastructure is SCADA software designed to control an electrical grid. For this purpose, we used a commercial SCADA product obtained through special research funding. The MTU and historian were hosted on a Red-Hat Enterprise Linux machine running the DNP3 version of the GENe SCADA software from General Electric [114]. The DNP3 version was chosen because of the popularity of this communication protocol for electrical grid ICS. The physical server provided by the software vendor was backed up and restored on a virtual machine. This impacted its performance, but it provided the ability to make snapshots of the machine for quick restoration. This trade-off proved critical for fast re-initialization of the experimental setup in a training setting and for saving development time.

The second piece of commercial software is the RTU emulator. The RTU Load Simulator (RLS) is special-purpose RTU software designed to perform load simulation for acceptance testing of GENe software. The RLS software is run on a virtualized Windows XP machine. Because each RLS typically represents an electrical substation, the RLS VM is cloned multiple times to achieve the desired scale. With the RLS machines (playing the role of RTUs) and the MTU machine, we have a fully functioning commercial grade implementation of a SCADA network. We can also add additional machines, such as operator workstations to enrich the network model. This implementation generates high fidelity traffic on the network. The implementation also responds exactly as a real system to cyber attacks. However, we have to address the physical component feedback. Because we consider the physical component to be a black box, we need to provide the

inputs from the SCADA system (i.e. the values of the command points) and integrate the outputs (i.e. the value of the measurement points).

The RLS behaves exactly like a GE RTU in terms of network communications, but does not interface with actual PLCs. Instead, each RLS has a database residing in RAM with values for each PLC. A command-line interface (CLI) was built by Rosset [115] to interact programmatically with the values. This was achieved by injecting a DLL into the RLS program memory to be able to read and modify values directly in RAM. A scheduled task on the RLS machine runs a script periodically to extract the values of the control points (through the CLI get method), and feed them to the power flow simulator and retrieve the results. The results are then fed back to the RTU emulator through the CLI set method. The frequency of execution of this script depends on the polling rate of the MTU and the convergence time of the electrical network simulator. The script needs to run faster than the polling to present accurate measurement point values, but must allow enough time for the simulator to converge.

Because of the bulkiness of physical equipment and because our scope does not require detailed granular fidelity of the electrical side, we chose not to emulate physical equipment. However, we still need a system that could provide us with the physical feedback a real system would present. We chose to use a power flow simulation to provide us with the physical feedback. The simulator requires a global knowledge of the state of the system. As such, it was more convenient to run a centralized simulation rather than a distributed one. In order to collect information about the state of the system and to update the local state of the RTUs, TCP requests are used. To enable this functionality, our architecture assumes a TCP server is running on the simulation server. Should this not be the case, one has to be built.

The experiment network is designed to connect to the electrical power flow simulator. This simulator may be hosted in the cluster for high threat experiments or hosted on a separate computing cluster. Should the power simulator be hosted on a remote network, a firewall would separate the experiment network from the power flow computing cluster for a number of reasons. The first reason is to prevent any traffic from the computing cluster to interfere with the experiment. At the same time, we do not want malicious software used in our experiments to contaminate the computing cluster. Thus, the firewall prevents all traffic from getting in and only allows the correct TPC requests to the simulation servers to get through. In both cases, soundness

of network captures is impacted by the communication with the electrical simulator because the out-of-band communication (TCP requests going to the simulator) use the experiment network. However, this can be addressed by filtering out this specific traffic either at the capture or post-hoc in the PCAP files.

When choosing a power simulator to integrate, we have to be conscious of the requirements of the experiment as the physical component will impact the degree of fidelity of the physical results, but also of the network traffic. For example, a very fine grained simulator that incorporates interference and noise in the power will produce measurements that vary more often than if a simple steady-state power simulator is used. In turn, this will trigger the SCADA system to report more updates, changing the traffic profile. Of course, the fidelity of the physical component is even more important for experiments that focus on impacts of cyber attacks on the physical side. For example, an experiment aiming to test the effectiveness of certain physical attacks in causing a spike in voltage that would burn out a specific piece of equipment, a reproduction of the AURORA experiment [26] for example, would require a very detailed model of transient effects in the power grid in real time, a firm model of physical protection mechanisms, an implementation of automated power grid operation and so on. A less detailed impact analysis, focusing on macro effects, might make due with a steady-state power flow simulator where end state values accurately reflect reality, but transient values, which have no lasting effect on power delivery unless they cause failures, are ignored.

There is an explicit trade-off for the electrical simulator between the complexity of the model and the granularity of the results. A more complex model will provide better granularity of results, for example a complex model might more accurately model transient effects. However, the complexity of the model might hinder scalability. For example, providing a real-time representation of transient effects might require a very detailed model of all the physical equipment used in the electric grid. The modeling effort involved in standing up an experiment in the scale of the entire grid is intensive. Additionally, the computing power required to provide results with this many components cannot be ignored. So, while the fidelity of results is impacted both on the physical side and on the network side, if sacrifices are made on the electrical simulator, the loss of fidelity may be acceptable when weighed in against gains in experiment setup time and computing power required.

To allow for the selection of a simulator of the appropriate type for each experiment, a modular approach was taken. Both the electrical simulator and the SCADA software were considered black boxes connected through a shim layer that runs a basic update logic. Using "set" and "get" methods in interfaces designed specifically for the software used, the update script makes sure that the values on both the SCADA side and the physical side are consistent with each other. This guarantees that any "set" (operate) request coming from the network over DNP3.0 is propagated to the electrical simulator and any "get" (read) request provides the most up to date data on the state of the network. This modular design enables the electrical simulator to be changed without changing the experimental design. Figure 4-2 illustrates the interaction between the SCADA module, the update script and the simulator module. In that figure, the SCADA black box represents proprietary components we acquired. Similarly, the electrical simulator portion ideally leverages existing technology. The update script, the interface with the simulator and the interface to the proprietary software each had to be built. More details about the implementations can be found in Rosset [115] for the RLSinjector interface and in section 4.3.3 for an example of an interface with the simulator.

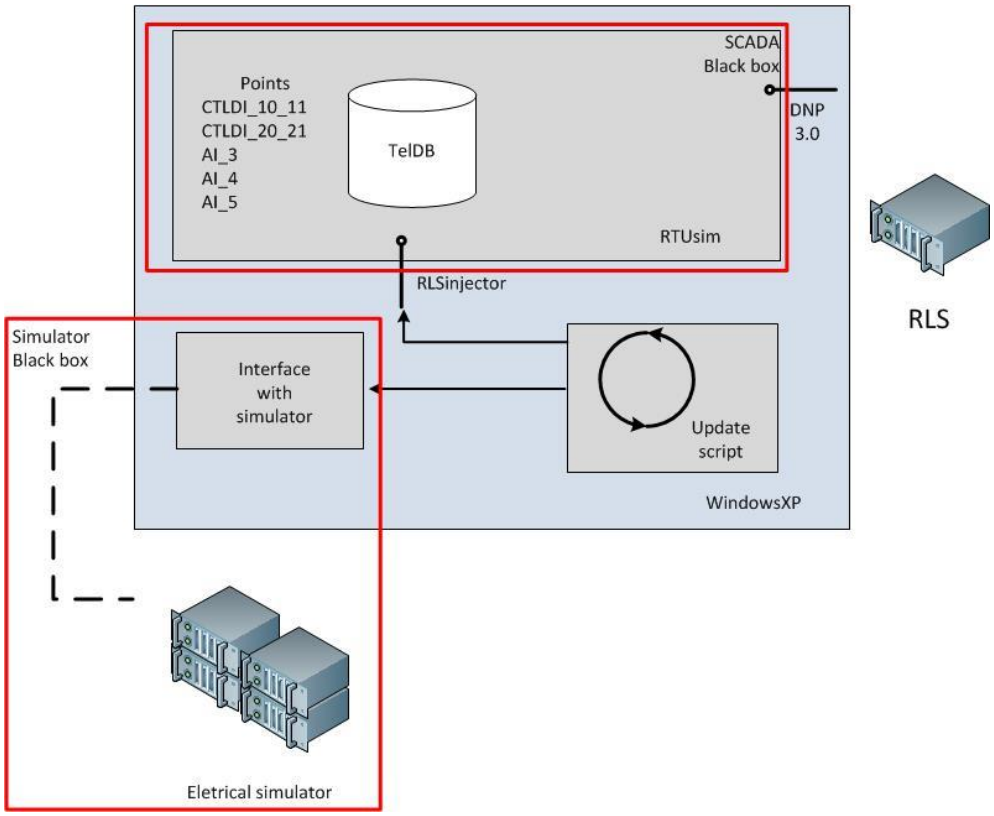

Figure 4-2: Black box design of simulator and SCADA modules

This black box design also makes it easier to change the simulator if another power flow simulator is used instead or if a different physical system is modelled. For example, if we wanted to model an oil and gas pipeline SCADA system instead of the electric grid, we could exchange the physical system black box.

4.2 Validation with the ICS sandbox SCADA emulation component

Because the ICS sandbox represents a novel approach for experimentation in cyber-physical systems, we must provide support for the validity of the approach. The easiest way to do so would be to reproduce results obtained on a physical deployment, but there is no such results in the public domain. So, in order to provide support for our approach, validation experiments are performed for each component, except for the emulated SCADA software, which is actual software used in production systems, and the electrical simulator, which is validated by the appropriate power engineering community. This section focuses on the validity of the emulation component and its ability to accurately represent an actual SCADA deployment.

The ICS sandbox had an opportunity to prove itself in training offered to industry practitioners. Due to logistical constraints of moving equipment to the training venue, it was not possible to move or remotely access the electrical simulator. However, the acknowledgement of usefulness of the emulation part of our approach from operators of real SCADA networks can provide some validation of significant parts of our work.

4.2.1 Description of the training

The training was organized by Natural Resources Canada (NRCan) which is the designated lead for energy infrastructure protection, including cyber threats. The training took place at the National Energy Infrastructure Test Centre (NEITC) located in Ottawa. The ICS sandbox was moved to that location for the duration of the training. A previous 1-day demonstration training on the ICS sandbox had been made to industry leaders in order to get their feedback on the type of training session that would be most valuable to their staff. The topic of incident handling in an ICS environment was identified as being the most important topic to cover. Consequently, introductory training on cyber incident handling in an ICS environment with a focus on hands-on interaction was prepared and delivered.

The training was conducted for 28 industry practitioners. The level of skill varied. Trainees included system administrators, SCADA system engineers, security experts, security policy practitioners, security managers, compliance consultants and penetration testers. All were working in industry, either for energy providers or for consulting firms working with them. The length of training was two and a half days. The ICS sandbox was used in four 90-min tracks and in a 3-hour training exercise on the last day. For the purpose of the training, additional machines representing corporate infrastructure were added to the ICS sandbox. The network infrastructure is presented in Figure 4-3.

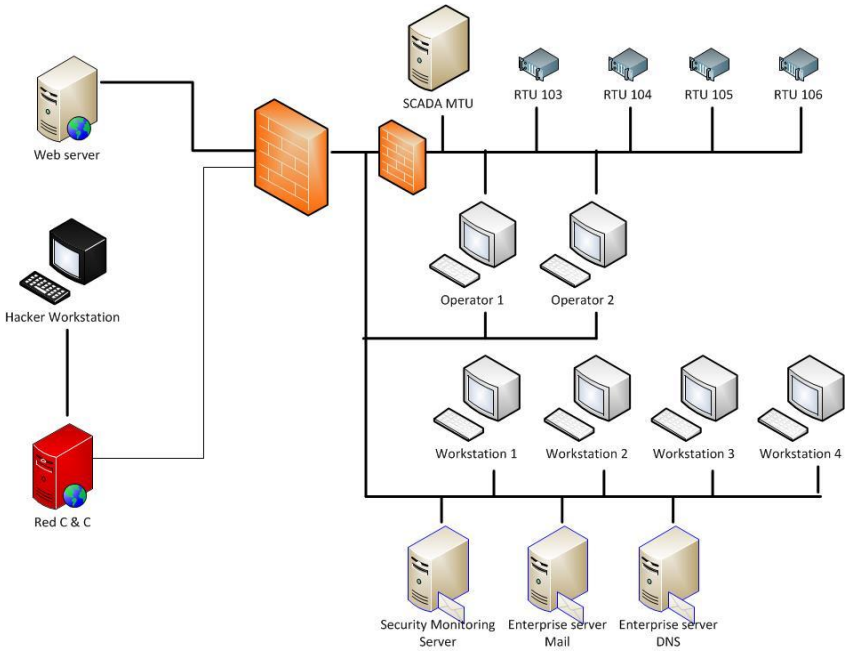

Figure 4-3: Training network infrastructure

The training configuration consisted of the ICS sandbox with one MTU and 4 RTUs connected to 4 PLCs each. We also had two "dual-homed" operator workstations (with one network card on the office network and another on the SCADA network) configured as HMI stations and four corporate Windows XP user workstations. Three servers provided enterprise services including mail, Domain Name Service (DNS) and security monitoring (Snort IDS). A small representation of the Internet containing one web server, one hacker workstation (Backtrack 5 R3) and a Web server for malware command and control was also included. A single OpenBSD machine was doing the role of router and firewall. A managed switch with VLAN support provided the layer 2 connectivity. All the machines were virtualized for easy restoration. The MTU and IDS were each running on a dedicated server and everything else was run on 3 desktop PCs with multiple

network cards. A dedicated control network to access VMware applications on each machine is not shown in the figure. The power flow simulator could not be physically moved and was not integrated in the training scenario.

The first track was a demonstration of how a persistent attacker would infect a machine in the office network then pivot and worm his way to the SCADA network through dual-homed machines. The second track was a demonstration of network components and counter-measures used in ICS, such as, looking at IDS and firewall logs and performing containment with the firewall. The third and fourth track were training on Wireshark and Sysinternal tools where a copy of one of the workstations was attacked by a drive-by download (automatically generated by the Social Engineering Toolkit (SET) Metasploit plug-in available on Backtrack) and numerous post exploitation actions were taken. The traffic from this attack was recorded and provided on the virtual image distributed to the students. The last-day exercise required the students to perform the full PICERL (Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned) incident response steps on the network shown in Figure 4-3. In the exercise scenario, we unleashed a custom-made program than emulated a worm. The initial infection was via USB key and the worm then connected to the external Internet command and control server and propagated over the network by brute forcing weak Windows share passwords.

4.2.2 Evaluation and lessons learned

Trainees were asked by the NEITC to fill out a questionnaire to help guide future training. In particular, they were asked to rate the course and the various sessions. They could give a grade of "adequate", "good" or "very good". Overall, the training was highly rated with 45% "very good", 55% "good", and 0% "adequate". In addition, all participants unanimously responded that they would recommend this course to a colleague.

Of the four sessions using the ICS Sandbox, two of them were very highly rated by the trainees. The SysInternals training track received 56% "very good" ratings and 33% of "good" ratings, the APT demo track got 40% "very good" and 50% "good". Participants were also asked which session they enjoyed the most. The most popular ICS-related session was the advanced persistent threat demo session (20%), followed by SysInternals tool workshop (16%) and the Wireshark workshop (13%). The sessions with the least amount of hands-on training finished last. This data seems to suggest that the ICS sandbox provided value to the trainees. It seems clear that the

students preferred hands-on exercise to lectures. The use of the ICS sandbox to generate materials for the exercise helped frame the hands-on in the context of industry practitioners

Additional conclusions can be taken from the observation of the trainees. During the hands-on sessions, a small majority of the students seemed to have good working knowledge of the tools covered in the hands-on sessions, but were interested in the exercise nonetheless, because they did not know that their knowledge of the tool could be relevant in the context of incidence response in ICS. For example, they knew Wireshark can be used to examine network traffic, but they did not know what traffic related to an incident would look like. They were able to observe artifacts of real attacks, something they cannot normally do on their production network. Trainees also learned when it was appropriate and effective to use the tools. This proved more relevant than how to use the tools for many students with prior knowledge. From this perspective, the ability of the ICS sandbox to perform and observe real attacks and provide before/after pictures of infected systems, probably proved to be a key factor in achieving the high satisfaction results we observed across a wide range of attendee skill level.

4.3 Validation of the ICS sandbox simulation/emulation approach

Because the ICS sandbox represents a novel approach for experimentation in cyber-physical systems, we must provide support for the validity of the approach. The easiest way to do so would be to reproduce results obtained on a physical deployment, but there is no such results in the public domain. So, in order to provide support for our approach, validation experiments are performed for each component, except for the emulated SCADA software, which is actual software used in production system, and the electrical simulator, which is validated by the appropriate power engineering community. This section focuses on the validity of the hybrid simulation/emulation approach and its ability to be used in experimental research.

The ICS sandbox was used for training, but the training was not using an electrical simulator nor was it required to provide experimental results. As such, a second experiment, designed as a proof of concept for the hybrid emulation/simulation approach was realized. This section describes an experiment using the ICS sandbox to control the electrical network. First, the network that is the object of the experiment is presented, then the original experiment is reviewed. We continue with an explanation of how the ICS sandbox was configured for the experiment and we finish by presenting the results.

4.3.1 IEEE reliability test system

One of the main research thrusts in power systems engineering is increasing the reliability of power systems. However, there was no standard way of testing reliability schemes. So, in 1979, the IEEE Reliability Test System Task Force of the Application of Probability Methods Subcommittee presented a system that would address this lack: the IEEE reliability test system [116]. This system includes a load model, a generation system and a transmission network. Figure 4-4 presents the network. In other words the model includes the production network, the transmission network and the distribution network. The system is expressly designed to provide a variety of case scenarios, illustrating a range of production sources and a range of load types. Each of the production sources have different parameters in terms of capacity and production costs and each of the loads has a different load profile. The IEEE Reliability Test Task Force updated these values to reflect more recent profiles in 1999 [117].

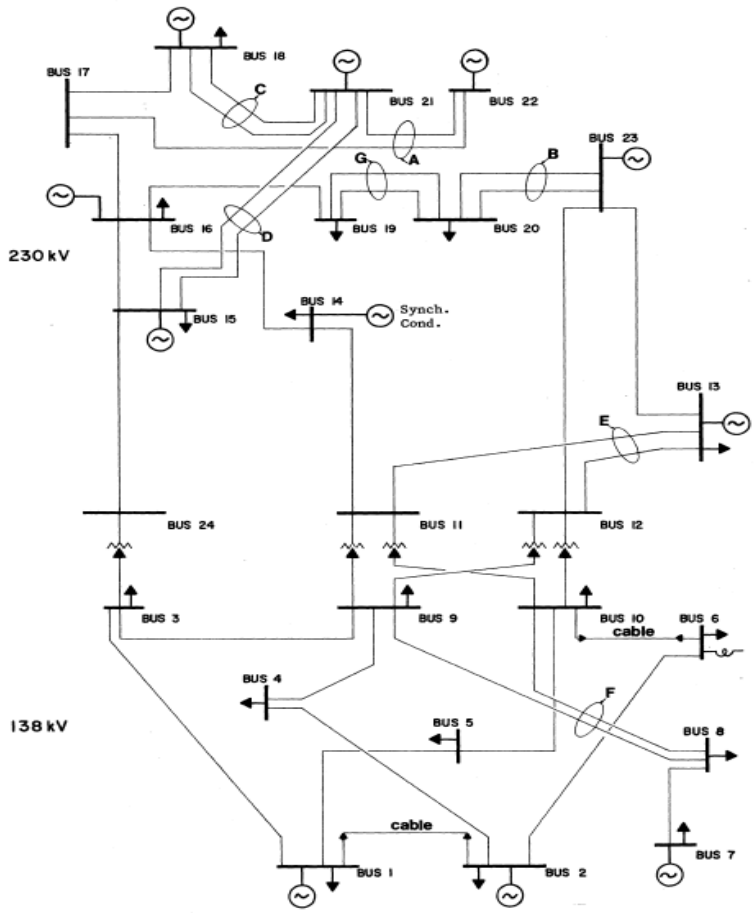

Figure 4-4: IEEE reliability test system network (reproduced from [116] © 1979 IEEE)

As illustrated in Figure 4-4, the network is comprised branches and buses. Each branch is an edge in the network and each bus is a node. So, the branches connect buses with each other and are denoted by the buses they are connected to. For example, the branch in the lower left corner of the figure, between Bus 1 and Bus 3, is called Branch 1-3. In the physical world, the branches would be power lines. Because power lines sometimes run in parallel, the reliability test network specifies, using circles and letters such as the circle A encircling branch 17-22 and branch 21-22, the power lines that are collocated and that will fail simultaneously.

If the branches are the power lines, the buses are the conductors in substations on which all production and distribution networks tap to provide or take power. Figure 4-5 provides a simplified illustration of a distribution bus.

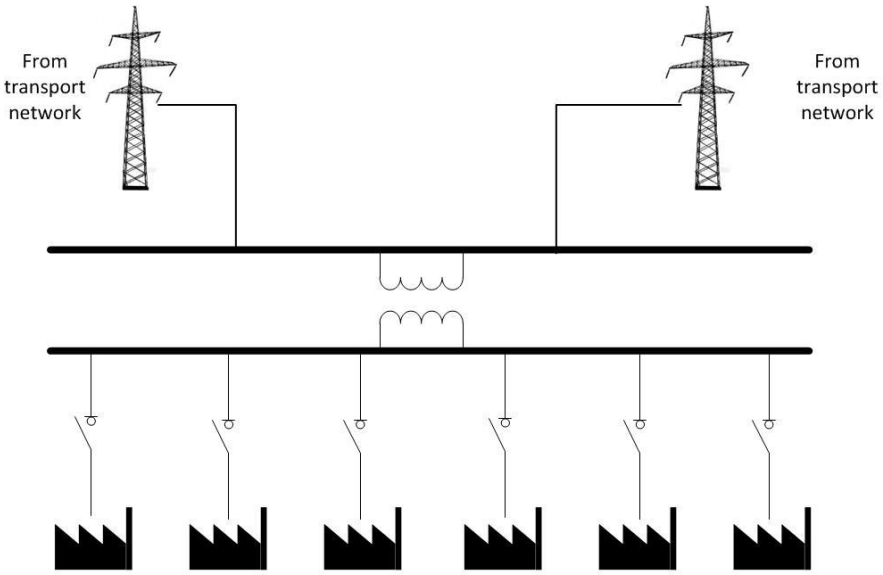

Figure 4-5: Simplified distribution bus

Power coming from the transmission network is put on a high voltage bus. A transformer then transforms the voltage to a low voltage level. The power then goes on a low voltage bus where distribution lines redistribute it to clients. . In that sense, all the loads form a parallel circuit. Alternatively, in the case of switching nodes, the power is instead transferred to another section of the transmission network Production buses follow a similar architecture. The buses in the reliability test system function in that way. Each bus acts as a node where power can be transferred from one edge to the next and where loads, identified by ground symbols in the

figure, may consume power and sources, identified by round symbols with a sine wave in figure 4-4, can add power to the system.

The test system is divided into two zones. A 138kV zone and a 230kV zone. The two groups of buses, bus 3-24 and bus 9 to 12 groups, isolate each zone from the other. Each of these groups represent buses and branches that would be collocated in a substation.

Optimized power flow problem

The test system network description only lists the parameters of each piece of equipment. If we want to determine how power flows through the network, and what values the current and voltages phasors can be take on each element, we have to solve the power flow problem.

The power flow problem is defined as a numerical analysis tool aimed at analyzing the values in steady state of the forms of power, for example voltage, voltage angle, current, current angle, real power and reactive power. This analysis is typically done on line diagrams such as the diagram of the IEEE reliability test system in figure 4-4. In other words, starting from the one-line diagram, the power flow analysis attempts to find the power, voltage and current for all pieces of equipment. Table 4-1 illustrates the kinds of results that can be obtained from a power flow calculation.

Table 4-1: Power flow calculation example

Branch	From	To	From Bus Injection		To Bus Injection		Loss (I^2 * Z)	
#	Bus	Bus	P (MW)	Q (MVAr)	P (MW)	Q (MVAr)	P (MW)	Q (MVAr)
0	1	2	11.94	-26.92	-11.94	-22.45	0.004	0.02
1	1	3	-7.97	21.57	8.31	-26.11	0.342	1.32
2	1	5	60.03	4.83	-59.29	-4.37	0.741	2.87
3	2	4	38.44	19.15	-37.85	-20.43	0.587	2.27
4	2	6	48.5	-1.04	-47.41	-0.19	1.093	4.22
5	3	9	22.9	-17.01	-22.66	14.75	0.24	0.93
6	3	24	-211.21	6.12	212.32	34.48	1.113	40.6
7	4	9	-36.15	5.43	36.52	-6.83	0.364	1.41
8	5	10	-11.71	-9.63	11.76	7.3	0.046	0.18
9	6	10	-88.59	-130.31	89.66	-121.12	1.067	4.64
10	7	8	115	26.84	-112.88	-20.35	2.118	8.18
…	…	…	…	…	…	…	…	…

Additional information, such as the power distribution in buses and the voltage and current for branches is also available. This is considered the *base case* for power flow analysis.

To solve this base case, the numerical solution must follow a certain number of constraints. In his book section [118], Bacher presents a summary of the constraints needed to build a mathematical model for the simulation. Essentially, the following physical constraints must be met :

- Energy conservation in passive power elements
- Kirchoff's law of currents (the sum of all currents in a node must be equal to zero)
- Ohm's law (power-voltage-current relationship) for all elements

These constraints ensure that all power generated by the source eventually makes its way to ground through a load. The constraints also ensure that the current and voltage values in intermediary elements represent the physical behaviour of the electric grid. Additional constraints based on physically enforced operating tolerances for the equipment may also be enforced. for example a line may not exceed its base operating voltage by more than 5%. If it does, it will trigger a physical protection mechanism that will shut down the line. So, while transient effects may trigger open lines, it is impossible to observe these values in steady state. Thus, we must create constraints to prevent these values from appearing in the solution space.

For the majority of systems, there is more than one solution that meets all the constraints. For example, in the updated IEEE reliability test system, there is about 20% excess generation capacity compared to the total load. This means that there is a number of generation configuration that can meet demand. In order to satisfy the power current relationship, the loads, defined as power consumption, must induce a specific amount of current. In order to satisfy Kirchoff's law of currents, some sources must be turned off. The base case power flow analysis does not discriminate between the solutions and returns a numerical solution that fits the constraints. The *optimal power flow* analysis finds the solution which meets the constraints at the lowest cost. The costs are calculated based on parameters provided by the operator. For example, in the case of the IEEE reliability test system, the costs is calculated from generation parameters attributing to each source a cost per unit of power produced based on the type of power plant it emulates.

4.3.2 Terrorist threat experiment

The 9/11 terrorist attacks spurred a large volume of research in defending against terrorist attacks. In particular, the defense of the electric grid was identified as a point of vulnerability. In 2004, Salmeron et al. [119] analyzed the resiliency of the electric grid to terrorist attacks.

The goal of their paper was to identify critical components of the electric grid by evaluating how terrorists could maximize their damage with a given set of resources. In their words, they strive to *identify critical sets of a power grid's components [...] by identifying maximally disruptive, coordinated (nearly simultaneous) attacks [...] which a terrorist group might undertake* [119]. Because they are uncertain of what kind of resources a terrorist group might possess, they consider a range of capabilities. However, they only consider physical destruction and assume that it is impossible for the group to perform cyber attacks on the SCADA system.

To perform their study, Salmeron et al. use the 1996 IEEE reliability test system, the test system with the revised values presented in section 4.3.1. Using the reliability test system parameters, they construct a direct current optimal power flow model (an approximation of the actual optimal power flow model) DC-OPF. This model contains the usual constraints, but adds constraints to be able to shed load if there is not enough power generation resources available, notably that load shedding S_{ic} cannot exceed demand. Then, they create disruption by removing interdicted components based on terrorist capabilities. For example, if a terrorist would blow up a pylon, all lines attached to it would be turned off. Once these components are removed, a new DC-OPF is calculated. A new function, I-DC-OPF, maximizes the impact of the interdiction on the power flow of the system. The interdicted components of the solution of I-DC-OPF form the terrorist interdiction plan.

Among their findings, Salmeron and al. identify two interdiction plans for the single IEEE reliability test system. These "near-best" plans, are reproduced in Figure 4-6. In the first plan, the main substation, interconnecting buses 9, 10, 11 and 12 is destroyed and a number of lines (both lines of branch 15-21, branch 16-17 and both lines of branch 20-23) are cut. In the second plan, only lines are cut (branch 7-8, branch 11-13, branch 12-13, branch 12-23, both lines of branch 15-21, branch 16-17 and both lines of branch 20-23). Of these two plans, plan 2 sheds slightly more load (1373 MW compared to 1258 MW), but plan 1 is identified as being the most severe. This analysis is based on the destruction of the substation in plan 1 which is dubbed more

difficult to repair than the line cuts in plan 2. The reasoning is that the cost in the entirety of the outage, measured in MW·h, will be much higher if the impact in power is similar, but the time to repair is orders of magnitude larger.

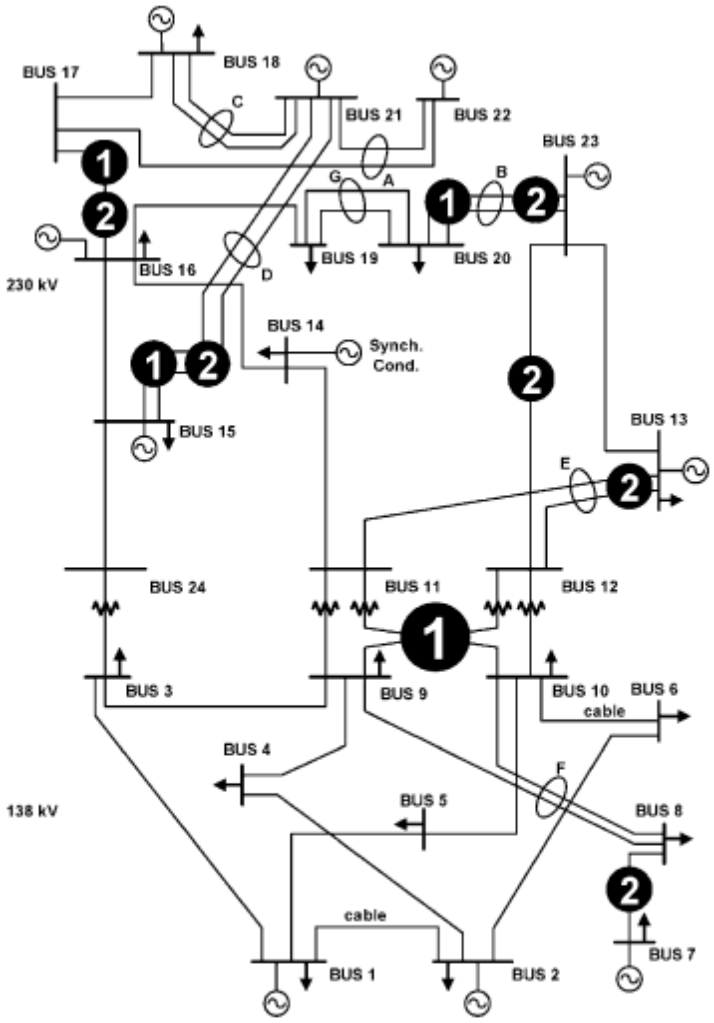

Fig. 3. Two interdiction plans (depicted as ① and ②) for RTS1 using $M = 6$. Total load is 2850 MW. Plan 1 sheds 1258 MW and plan 2 sheds 1373 MW. The large "①" indicates that the four transformers and buses in the substation are interdicted.

Figure 4-6: Near-best interdiction plans proposed by Salmeron et al. (reproduced from [119] © 2004 IEEE)

4.3.3 Adapting the ICS sandbox

In their paper, Salmeron et al. have produced a consequence-based analysis of the impact of physical terrorism. They assumed cyber attacks would not be possible. However, if we could

replicate their physical attacks using cyber components we could perform similar impact assessments for cyber attacks. In order to do so, we must configure the ICS sandbox to fit with Salmeron et al.'s experiment.

Electrical simulator

The is a requirement to find an electrical simulator that can perform power flow calculations and can fit in our black box model. We can either solve the *base case* of the power flow analysis or opt for *optimal power flow*. There are advantages and drawbacks for each, so we must carefully weigh the options. If we solve the *base case*, we do not have any control over the numerical solution used. While the solver is likely to always produce the same solution for identical network states because the model is not probabilistic, the solution may be one of many. There is no guarantee that a utility operator would select that particular solution instead of one of the others. In fact, some solutions might actively be avoided by utility operators because of their cost. On the other hand, that same fact represents one of the advantages of using the *base case*. The lack of evaluation of the fitness of solutions does not make any assumptions about the behaviour of the utility operator and about the information at his disposal. The reverse is true for *optimal power flow*. Any rational utility operator would operate his network in order to minimize production costs. So, if an operator has a power generation discipline, it is highly likely that he will produce generation choices similar to the results of the optimal power flow calculations. In fact, many automated power generation algorithms use optimal power flow calculations to regulate power. Unfortunately, to use this discipline, we have to assume that the utility operator possesses all the knowledge required to perform this calculation. Notably, a good estimation of the state of the network and of the load is required. This is unlikely to happen if the attacker is willing to falsify the data returned by the SCADA network used in those calculations.

If the scope of the experiment is to reproduce the attacks from Salmeron and al., the attackers are only interested in shutting down the system to maximize interdiction. They only require the capability of shutting down the system using SCADA commands. This capability requirement is much less severe than the ability to send arbitrary grid state evaluation to the grid operator. Ultimately, as shown in Chapter 3, this is not the scenario an advanced attacker is likely to attempt, but its study would still have merit to model the effect of a destructive attack like Shamoon. In this kind of scenario, the production network operator is likely to have a good state

evaluation and continue to apply his discipline. Another particularity of the interdiction scenarios presented is that they only affect portions of the transport network: a substation and lines. As presented in section 2.1.3, the production network, distribution network and transport network are often controlled independently, as islands. By focusing our attention on the transport network SCADA system, we can assume that an attacker, even if he fully infiltrated the transport network, requires a different attack to compromise the production network's control system. In that case, the production network could still function optimally with no restraints to the attacker's capabilities. For these reasons, we should use an *optimal power flow* solver.

PyPower [120] is a Python port of MATPOWER, a Matlab power simulation package. This simulation package is able to solve power flow and optimal power flow problems. It is also possible to describe line diagrams of electrical networks in matrix forms. It can also take into account costs for optimal power flow calculations. In addition, PyPower has a native description of the 1996 IEEE reliability test system case. The convergence time is relatively fast, in the order of seconds, which is fast enough for our update script. In addition, optimal power flow calculations provide an estimation of the generation costs based on the 1996 IEEE data. On the downside, the optimal power flow calculator strictly enforces constraints and load shedding is not allowed. This means that, under severe disruption, the solver may not converge on a solution that satisfies all the constraints. In that case, the solver will produce a solution that follows the three basic physical constraints (conservation of energy, Kirchoff's law of currents and Ohm's law), but may violate operating constraints for equipment.

PyPower does not have a native network interface. However, because it is based on Python, we can create our own. We build a multithread TCP server that will be able to serve all the RLS at the same time if required. The server receives a communication from a RLS that contains the values of the control points and the name of the RTU. Once the message is received, the server looks in a correspondence table that matches the name of the RTU and the values received with pieces of equipment in the IEEE reliability test system description. The state of the test system is updated with the values of the control points. For example, if the breaker for the branch 15-17 is opened, the status value of the branch 15-17 line is set to zero. The PyPower simulation preserves the state of the system to make it available to all RTUs, then runs the optimized power flow calculation to calculate the measurement points and the generation cost for the computation. The results of the computation, the time of the simulation and the generation cost are stored locally.

Then, the server, based on a correspondence table, reads the power flow results and sends the value of the measurement points via the TCP response. The architecture is summarized in figure 4-7.

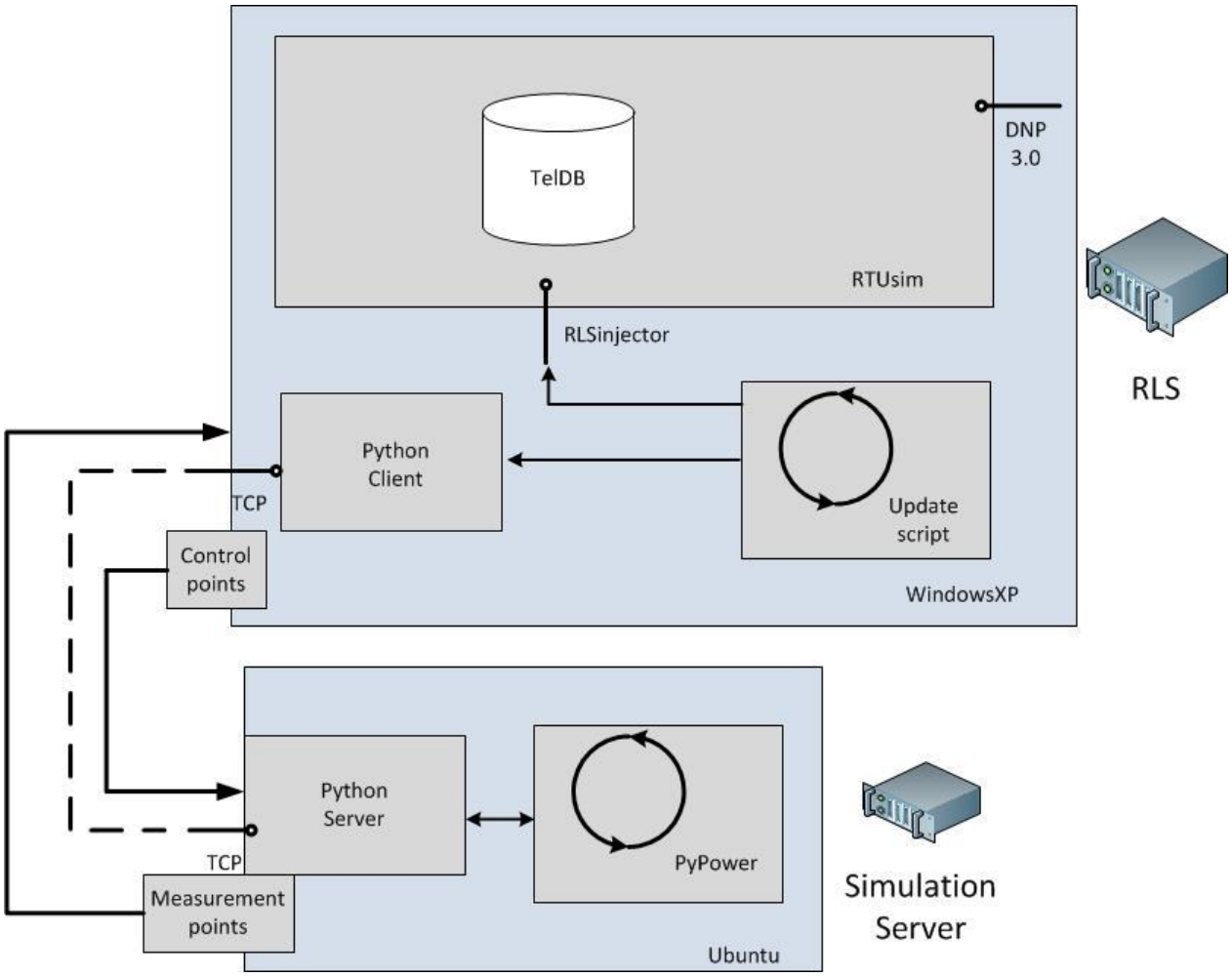

Figure 4-7: Black box architecture using PyPower as power simulation

SCADA network

The SCADA network needs to be customized to fit the IEEE reliability test case. In particular, we need to implement a SCADA control scheme that will enable us to replicate the interdiction scenario.

There is no public domain documentation of to the extent to which operators integrate SCADA to their operations. The IEEE reliability test system is not an exception to this rule. So, in order to limit the experience setup time, we will limit the amount of control to the minimum required to replicate the scenario. Adding additional RTUs or control points is possible, but requires manual

configuration of the SCADA software which is time consuming. So, we will opt for a configuration with four RTUs : one RTU as the bus 9 to 12 substation, one RTU on bus 20, one RTU on bus 17 and one RTU on bus 15. The substation RTU controls the lines connected to the transformers. The RTUs on the various buses control the lines connected to that bus. In that sense, the bus RTUs represent the transport network side of a transmission to distribution substation. In addition to controlling lines, each RTU reports the voltage value for each bus in its area.

In terms of control points, each RTU has one 1 bit digital control point for each branch. For example, the bus 17 RTU has the following control points :

- BRANCH-16-17
- BRANCH-17-18
- BRANCH-17-22

Each control point has a default value of ATP_CLOSE, meaning that the breakers are closed, and current is allowed to go through, by default. Should the value change to ATP_OPEN, the status of the line has to be changed to 0. This is done by the PyPower server by changing the value of the power flow data structure. The power flow data structure that holds the IEEE reliability test system includes the "branch" array that is an array of branch type structures. The eleventh value of the branch type is status which is a binary value. Should that binary value be changed to 0, the branch is removed from the line diagram.

In terms of measurement points, each RTU has one Analog point for each bus. The analog point records the value of the amplitude of the voltage of the bus. For example the bus 17 RTU has the following measurement point :

- BUS_17

This measurement point cannot be used to perform control and only records the value obtained from the electrical simulator. Once the PyPower server finishes calculating the optimal power flow, the voltage amplitude can be accessed from the "bus" array of the power flow data structure. The eighth value of the bus structure is the amplitude of the voltage as a fraction of the base voltage. The tenth value is the base voltage of the bus. The actual voltage value can be obtained by multiplying these two values.

Once all RTUs are configured, we can now control the transport side of the IEEE reliability test system and receive feedback and report the state of the electrical network centrally.

4.3.4 Reproducing the scenario

With the ICS sandbox configured to match the IEEE reliability test system, we only have to operate the SCADA controls to affect the state of the test system. The effects of the scenario from Salmerin et al. can now be reproduced with cyber attacks.

Unfortunately, while the effects can be reproduced, it is not possible with the current simulator to reproduce the results. Salmeron et al. report the results of their maximally disruptive attacks in terms of amount of load shed. As we have seen in section 4.3.3, the PyPower simulator does not allow load shedding. However, we can track the generation costs in real time and estimate the damage of the attacks in terms of increased generation cost. Because the damage is tracked in real time, we can also see the effect of each interdiction as it happens, allowing us to evaluate the impact of each interdiction separately. It would also be possible to perform all the interdictions at the same time. It was deemed preferable to allow for a delay between each interdiction to see the individual effects. This delay, however long in the scale of cyber attack, is negligible compared to the ability of even the best terrorists to coordinate physical attacks.

In theory, the order of the interdiction influences the individual effect of an interdiction. For example, a break in a line might have little effect if the grid is in a relatively stable state. However, that same break might have disastrous consequences if the grid is already overloaded from previous failures. In practice, because the impact of individual interdictions in this experiment is only provided in a proof of concept framework, the validity of those impact has little bearing on the results. So, we adopt the following arbitrary ordering of interdictions:

1. Interdiction of the transformer in substation 9 to 12
2. Interdiction of both lines from branch 15-21
3. Interdiction of the line from branch 16-17
4. Interdiction of both lines from branch 20-23

The generation cost of the optimal power flow in the face of these interdictions is presented in Figure 4-8.

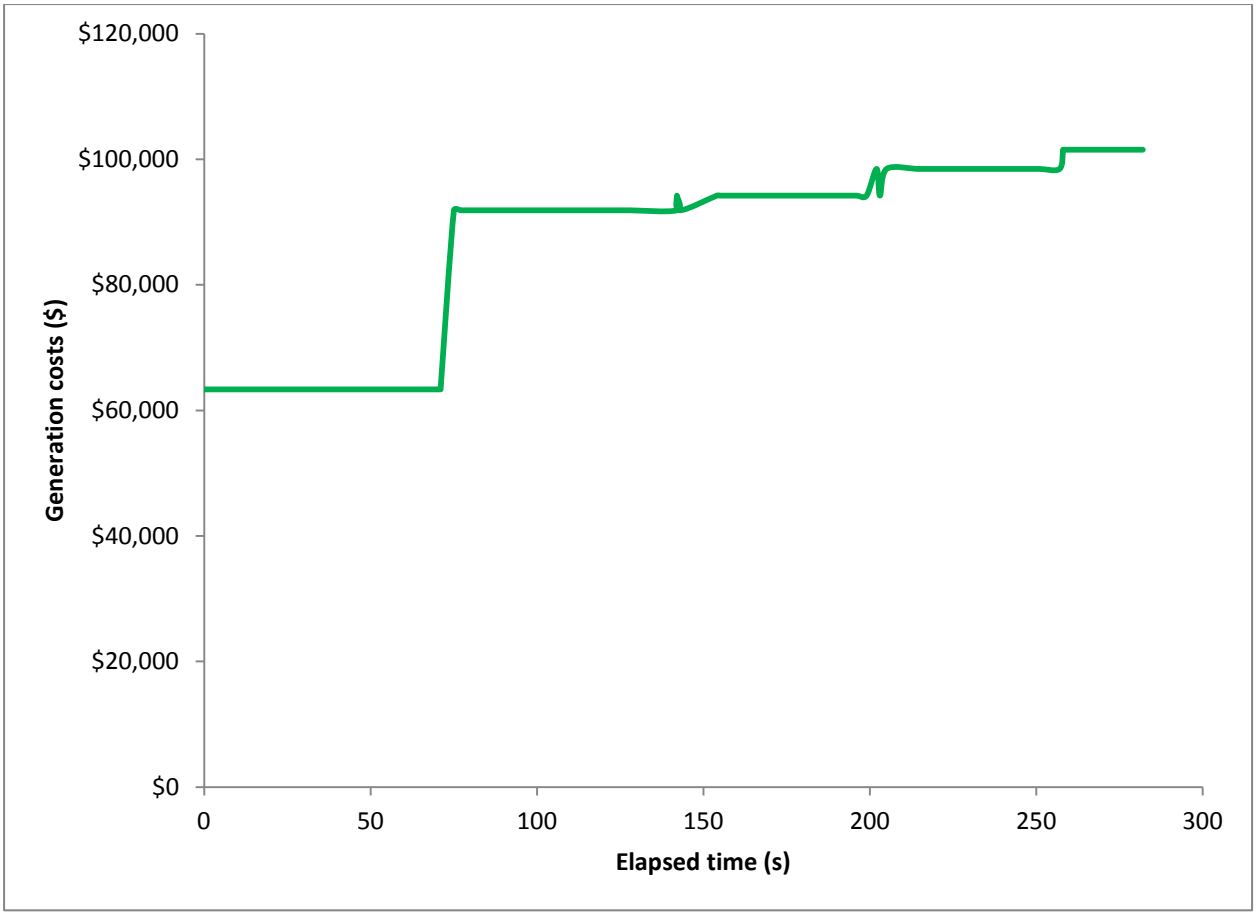

Figure 4-8: Impact of the interdictions from scenario 1 on generation cost

The effects of the loss of the transformer at around 75s can be clearly seen, imposing a 30 000$ burden on generation costs. The loss of the 15-21 branches around 150 has a smaller impact of around 3 000$ and further losses of branch 16-17 and branches 20-23 also produce impacts of similar magnitudes. Apart from the transients around the times of the interdictions due to the multithreaded nature of the server which may cause race conditions in the state of the system, the generation cost graph follows a strictly increasing cost curve as we would expect from mounting damages in the wake of a terrorist attack. This suggests that, should we use a simulator with the capacity to shed loads, it would be possible to reproduce Salmeron and al.'s experiment.

This proof of concept experiment showed that the emulation/simulation approach can be used to produce experimental results. Should additional research from power engineers provide us with a simulator with the required capabilities or with a model for the case study where load shedding priorities were determined, sound results could be achieved with limited modifications to the

implementation. This lends credence to the claim that the ICS sandbox hybrid emulation/simulation can be used to perform cyber security experimentation.

4.4 Conclusion

In this chapter, we have seen the ICS sandbox approach. This approach combines emulation of cyber components, to provide high fidelity network traffic, and simulation of the electrical components, to provide suitable feedback for the SCADA system at a reasonable cost and scalability. This enables us to generate network traffic that resembles the traffic of real SCADA networks at scale. This contrasts with current state of the art in experimental SCADA research where the focus is not put on high fidelity network traffic, the experimental network is not scalable or the cost of standing up an experiment is too high for most academic institutions. As such, the ICS sandbox approach represents a significant contribution to the community.

In order to lend credence to results obtained from the ICS sandbox, efforts were invested to validate the ICS sandbox. Because no publicly available data sets could be used to calibrate the sandbox and produce a validation experiment, each component was validated individually. The validity of the emulated SCADA software was not evaluated because actual production level SCADA software was used in the experiment. The traffic produced by these elements is the same as the traffic produced in a real SCADA network. Similarly, the electrical simulation was not evaluated because subject matter experts in power engineering can provide the validation for whatever electrical simulator and electrical model are selected for experimentation. So, validation efforts were focused on validating the design of the emulation approach to SCADA network components and on the interaction of emulation and simulation.

To test the ability of the emulated SCADA components, training sessions were conducted for members of industry. This training used the emulation component of the SCADA sandbox as a basis for the hands-on part of the training. Overall, the attendees were satisfied with the training in general but showed an even greater appreciation for courses with hands-on training on the ICS sandbox. Comments by users mentioned the ability to easily translate the hands-on training received into their own operational context and illustrate how the ICS sandbox successfully recreated an environment with which they were familiar, which is to say a production SCADA system.

For testing the emulation/simulation approach, the reproduction of a simulation only experiment conducted by power engineers in the ICS sandbox was performed. Using SCADA as a control element, the optimal interdiction scenarios proposed by Salmeron and al. were realized and the evolution of the production cost was recorded. While the actual values cannot be presented as results due to constraints in the power simulation software used, the ability to faithfully recreate the scenario in the ICS sandbox acts as a proof of concept of the emulation/simulation approach.

Using the novel approach of the ICS sandbox, it is now possible to generate high fidelity network traffic for SCADA security experimentation. For example, using a simulator that integrated a load shedding model, we could evaluate the impact of cyber terrorists in terms of energy production costs using the same framework we have used for creating our proof of concept. We can also focus purely on network security and perform experimentation leveraging the high fidelity of the network traffic. In particular, we can now test the proposed defensive strategy, which consists of making it difficult for attackers to use covert communications, in conditions resembling real SCADA networks and with real SCADA traffic.

CHAPTER 5 ANOMALY-BASED INTRUSION DETECTION IN SCADA NETWORKS

By studying the behaviour of advanced threats, we presented a strategy of attacking their ability to use covert communication to perform maintenance and expand their penetration in networks. The effectiveness of this strategy depends on the capabilities of the Warden to identify inmates communication with each other. In normal networks, where almost anything goes, it is difficult for the Warden to distinguish between unusual non-malicious traffic and malicious traffic. However, our intuition tells us that SCADA networks are unlike traditional corporate networks. Because of a polling based protocol, the traffic is well regimented and should provide a more regular backdrop against which malicious traffic can more easily be identified. In order to test this intuition, a new experimental method combining simulation and emulation allowed for the generation of high fidelity network traffic, which will serve as the data set for our experiment.

Using data from the ICS sandbox, we will be able to test the effectiveness of our defensive strategy. Under normal circumstances, it is difficult to construct a feature set that is suitable for use with anomaly-based detection. So, if, by choosing a few simple features and evaluating how these features differ from the baseline in the case of a compromise, we obtain effective anomaly-based intrusion detectors, we will know that anomaly-based intrusion detection performs better in SCADA networks than in the general case. In turn, this would lead credence to the foundation of the defensive strategy envisioned to defend SCADA networks against advanced persistent threats.

In this chapter, we present the results of an experiment in which the effectiveness of anomaly-based detection in a SCADA network is tested. Section 5.1 offers the methodology and experimental design used to characterize SCADA traffic. Section 5.2 presents the characterization of non malicious SCADA traffic according to three features, logical topology, interdeparture of packets and packet size, selected for their simplicity and good indication of compromise. Section 5.3 presents the three attack scenarios, common botnet, APT and covert channel, that were used as test cases for detection. Section 5.4. presents the results showing that even with the simple features, it was straightforward to detect most scenarios. The covert channel scenario proved more difficult to detect because of its high similitude with a regular source and is presented as the boundary for detection.

5.1 Methodology

Our analysis of the behaviour of APT has identified that targeting their ability to remain covert while they perform routine maintenance tasks could be used as a valid defensive strategy. Also, our intuition tells us that SCADA networks, unlike traditional corporate networks, behave in a much more deterministic way that could be leveraged to enhance the performance of anomaly-based detection. Common wisdom deems anomaly-based detection to have limited effectiveness. If we can create an effective anomaly detector for SCADA based on simple metrics, we will have proven the common wisdom wrong for the case of SCADA networks. This section presents a methodology to test the effectiveness of an anomaly-based detector for SCADA networks. First, a conceptual framework for characterizing traffic is presented. Then, the experimental setup used to perform the experiment is detailed.

5.1.1 Characterizing SCADA traffic

A number of authors [121], [122] have talked about the difficulty of modelling cyber-physical systems, such as SCADA networks. If we consider the case of the electric grid, there is still ongoing research into modelling both the grid and the control network in isolation. Studying them together is more rare. As a matter of fact, we can wonder if there is any impact of using a divide-and-conquer approach and studying each component of the cyber-physical separately.

Using the Shannon communication model, the physical component represents the source of the communication. The information that the source wants to communicate is the state of the network as represented in values of measurement points. For some measurement points, the value will seldom change. The example of the status of a protection circuit breaker comes to mind. Unless there is a failure in the grid, the value will stay the same. Other measurement points vary. For example, a meter measuring the voltage of a power line might be in constant flux based on the rigors of supply and demand on the grid. In that sense, the entropy of the source may vary from point to point and, ultimately, from system to system. The fluctuations directly affect the payload of SCADA packets.

Ultimately, the amount of fidelity in the representation of the physical part of the cyber physical system directly affects the source entropy of the communication. A complete abstraction of the physical system will leave a system with low entropy where it is easier to develop a number of

features for a NIDS. For example, if we have a system where none of the values ever change, the exact values could be used as a feature for intrusion detection. Obviously, this would allow a detector built on this data to appear overly efficient and closer to the ideal warden. If we look at the channel bandwidth in the presence of an ideal warden, we see that the bandwidth is limited to whatever entropy the protocol signaling adds if the entropy of the source comes closer to zero. As such, we must be wary of these effects when presenting research results.

Similarly, fidelity in terms of the representation of the cyber system affects the traffic characterization. In a sense, the cyber component represents the encoding, where measurements and control actions are given an electronic data representation, and it also represents the channel where the information is carried to the recipient. Because the cyber system is directly observed by the warden, it is critical that the bits and bytes of the network be as close as possible to traffic observed on a real network. If we use a NIDS, the warden reasons about the representation of the traffic on the wire. A change in the representation would inevitably distort the reasoning.

Because both cyber and physical components have an important role in the production of network traffic, the data needs to be generated by cyber-physical systems if we want a high level of fidelity. As detailed in section 2.4, current experimental approaches are not adequate to generate this kind of data. So, we must propose our own experimental approach to generate our dataset.

5.1.2 Experimental Setup

For our experiment, we want to generate traffic that resembles traffic from a live network. We will also want to have traffic that resembles real world threats. The easiest way to obtain the fidelity we need is to use real SCADA applications and real malware. So, we used the ICS Sandbox approach as described in section 4.1. For this particular experiment, we are not planning on evaluating the impact on the grid. This means, we do not need a high fidelity for the electrical stimulation. In fact, because we are not planning on doing deep packet inspection, the values output by the simulator are irrelevant. It is only important that the values are present. Also, we plan on using real malware. This makes the risk of using a remote simulator higher, so we will use a local simulator instead. Based on these requirements, we use an simplified electrical simulator that does not represent high fidelity scenarios, but that still manages to introduce some dynamism on the physical side, i.e. sending control messages will change the values reported by the RTU.

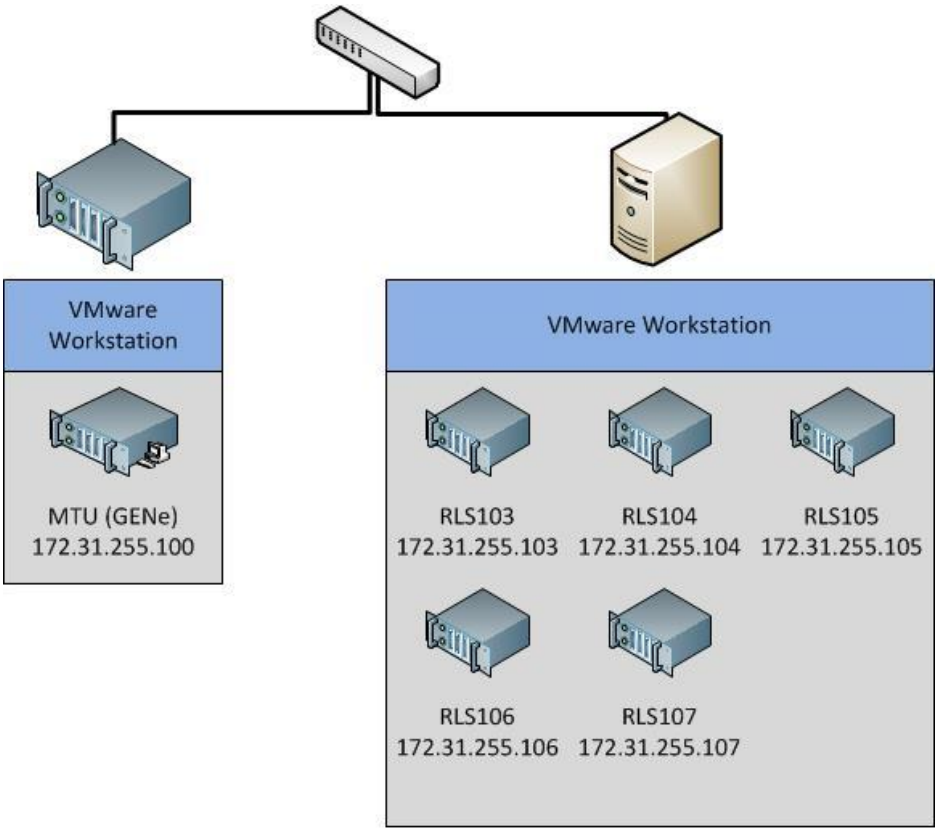

Figure 5-1 : Experiment network

To build our scenario, we based our design on the network used in the training described in section 4.2. We took the SCADA component of the training scenario as a baseline and added one RTU to become infected. The resulting network is shown in Figure 5-1. Looking at the figure, we see we have one MTU and five RTU. The MTU we use is a virtualized version the DNP 3.0 version of the General Electric GENe product [114] which was obtained through special purpose funding. Because of licensing issues, the MTU also serves as the HMI station. This slightly impacts our results as no HMI traffic will be seen on the network. However, our focus is the SCADA traffic, so the loss of HMI traffic is acceptable. We deployed six Windows XP machines running the RLS software to act as six RTUs. Each RTU is responsible for two control points and 3 measurements points for which the values are stored in a database accessed by the RTU simulator software. Experiments with a different number of RTUs and points were also run to evaluate the sensitivity of our results to variation in these control variables and results are shown in section 5.2.5. Based on this setup, full packet captures were taken by using port mirroring on the switch.

In order to integrate the physical components, in our case study an electrical grid, we wanted to make sure the values reported by the RTU integrated the dynamism inherent to cyber physical networks. To do so, each RLS machine was treated as a substation with one 12 kV main line supplying it with power and three distribution lines with static 1kΩ loads. The two control points were used as breakers for two of the distribution lines, allowing us to shut off power to two of the loads. Two measurement points reported on the current flowing through the lines we controlled with breakers. The last measurement point reported on the current going through the main 12kV line. A small, local, "electrical simulator" written in Python (sim_elec) implemented the electrical constraints imposed by this model. Figure 5-2 summarizes the design.

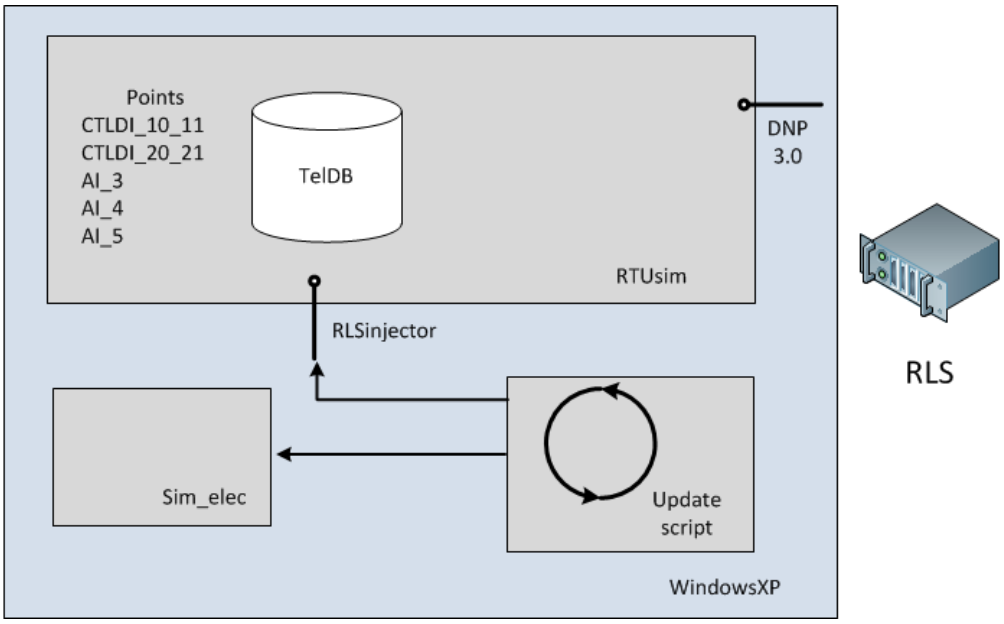

Figure 5-2 : Localized simulator design

5.2 A portrait of "normal"

Using the experimental setup, we can generate high fidelity network traffic. This traffic can be used to build a portrait of non-malicious traffic that will act as a baseline to spot anomalies. The first step is to select a number of features that are good representatives of the traffic and that can be used as indicators of infection. Then, we provide an analysis of non-malicious traffic for each of the three features, logical topology, interdeparture time and packet size, in turn. Finally, a sensitivity analysis of the impact of the experimental setup design choices in terms of number of RTUs and number of points per RTU is realized to ensure our design choices do not significantly affect the distribution of the features.

5.2.1 Feature selection

In order to build our detector, we need to find a good candidate for anomaly detection. Looking at network security literature, we can find a large volume of research that looks at characteristics of traffic to identify the underlying protocols or applications. The idea is to look for artefacts, in the form of indicators in the network traffic, that hint at the use of a specific protocol or application. For example, a particular cloud service might send synchronisation packets at regular intervals to make sure the state of the client is always good. A myriad of techniques can be used to do the classification. We can find examples of classification using statistical techniques [123], clustering algorithms [124], [125], Bayesian analysis [126], machine learning [127] and so on. The various approaches and the various selections of features that are used to uniquely identify the traffic have different strengths and weaknesses and are usually tailored to a specific use case such as finding encrypted traffic or performing quality of service decisions.

One possible set of features that can be used to classify traffic is packet size and interarrival time. Wright et al. [128] have shown that using only these features, it is still possible to obtain a reasonably good classification of a number of applications. This approach has the advantage of using only a small number of features, none of which require any deep packet inspection and protocol parsing. For SCADA protocols, such as DNP 3.0, not requiring protocol parsing is useful because, even though analyzers exist to interpret protocol headers, relevant information (e.g. is this breaker ON or OFF) is typically encoded and not readable without additional parsing and in-depth knowledge of the idiosyncrasies of each brand of equipment.

Previous work in characterizing worms using these measurements by Dainotti et al. [129] shows that a worm can be characterized by looking at interarrival time and packet size. Because worm traffic is generated by an automated process, the distribution of packet sizes and the time between the departure of two packets from the host differ greatly from those of traditional traffic. In SCADA networks, the requests for measurement updates by the MTU are also completely automated and have the potential to have similar properties which may be used to characterize the traffic. If the traffic is sufficiently characterized, we may be able to detect malicious traffic that falls outside the characterization.

From the packet captures, we select a subset of features we want to analyze. The first feature is the aggregate conversation flow characteristics, notably the IP source and IP destination pair. The

successful use of this characteristic to create network configuration and IDS signature by Hadeli [93] and Langill [105] suggest that this feature can be used to identify malicious behavior with traffic analysis. The second feature we select is packet size. This feature was a feature used to characterize worms by Dainotti et al. [129] and research by Wright et al. [128] indicated it, in conjunction with interarrival, time is a good classifier of network traffic. For the last feature, Dainotti and Wright both use time between packets, but one is receiver-oriented and the other is emitter-oriented. We select is interdeparture time, i.e. the time between the departure of two consecutive packets by the same host. This feature makes more sense with our polling mode of operation (in contrast with typical server architecture where clients initiate connections). The use of interdeparture also helps us observe cases where no responses are received, such as beaconing packets where no response is received.

Three methods are used to generate these features. For the conversation flow aggregate results, the packet captures are loaded into Wireshark and the conversation tool from the analysis toolset is used to generate a table of existing conversation pairs in the capture. The table is then exported using the copy function. For packet size, we use the Tshark tool, the Wireshark command line tool, to read the packet capture with the -e option to extract the frame.len field from the packet capture. This gives us the size of the frame as observed on the wire. Because we are using the same layer 1 and layer 2 technology for all RLS, the same packet from two RLS will have the same frame length. For a real network where this would not be the case, it would be possible to strip layer 1 and layer 2 headers from this value, but it was not required in our case. We also use the Tshark tool with the -e option to generate the interdeparture feature. The frame.time_relative field is extracted along with the ip.src and ip.dst field. This gives us the source and destination IPs in addition to the time from the start of the capture at which the frame was observed on the wire. Packets are then sorted by source IP address and ordered according the observation time. The interdeparture time is then calculated by taking the difference in observation time between two consecutive packets. We can now analyze the features using a spreadsheet application such as Excel or a mathematical analysis tool such as Matlab. In addition, we can reference the original packet capture to explain situations observed in the features.

So, to build our baseline, we will focus on two features:

- Interdeparture time between packets going to a same destination

- Packet size

This will provide a good baseline that should be able to be used for the detection of simple attacks. Should an attack be impossible to detect using this baseline, we will look at an advanced feature, the packet payload entropy, to see if it enables detection or if the attack mimics the regular traffic sufficiently to evade these techniques.

5.2.2 Logical Topology

A first characteristic of SCADA systems that may be leveraged to detect intrusions is the logical topology created by the master-slave aspect of the protocol. Even in an IP environment, the DNP 3.0 protocol has legacy embedded link layer operations encapsulated in the payload. This means that the SCADA machines will only communicate with other SCADA equipment for which they are preconfigured. In our case, the MTU can only communicate with the 5 configured RTUs and each RTU can only communicate with the MTU. We analyzed the packet capture using Wireshark's prebuilt conversation analysis tool. Table 5-1 summarizes the results based on the network and addressing plan illustrated in Figure 5-1.

Table 5-1: Wireshark conversation analysis

IP source (A)	IP destination (B)	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Duration (s)
172.31.255.100	172.31.255.255	172	18362	0	0	124.2
172.31.255.100	172.31.255.103	242	19904	215	18454	126.3
172.31.255.100	172.31.255.105	243	19980	214	18427	126.1
172.31.255.100	172.31.255.107	258	21249	224	19681	126.1
172.31.255.100	172.31.255.104	241	19845	215	18468	123.3
172.31.255.100	172.31.255.106	237	19498	211	18164	123.2

As expected, the only conversations we can observe are between the MTU and the RTUs. No communication between RTUs exists. In addition, all the conversations have approximately the same duration, number of packets exchanged and number of bytes exchanged. This result is also

expected because the speed of the automated polling is superior to the speed with which a human can operate the controls through the human machine interface.

5.2.3 Interdeparture time

Another legacy aspect of the DNP 3.0 protocol is the concept of polling. While the protocol allows for unsolicited communication originating from the RTU in case of failures, the normal mode of operation is polling from the MTU. The MTU polls each of the RTUs in turn to update the values of the points for which the RTU is responsible. This means that, for a given RTU, the interarrival time of polling requests is approximately constant. Each polling request is then followed by a small number of responses (e.g. returning requesting measurements) and ACK (acknowledging MTU communication) packets sent in short succession and a confirmation packet is then sent. Figure 5-3 illustrates the average time between the departure of two packets from the MTU.

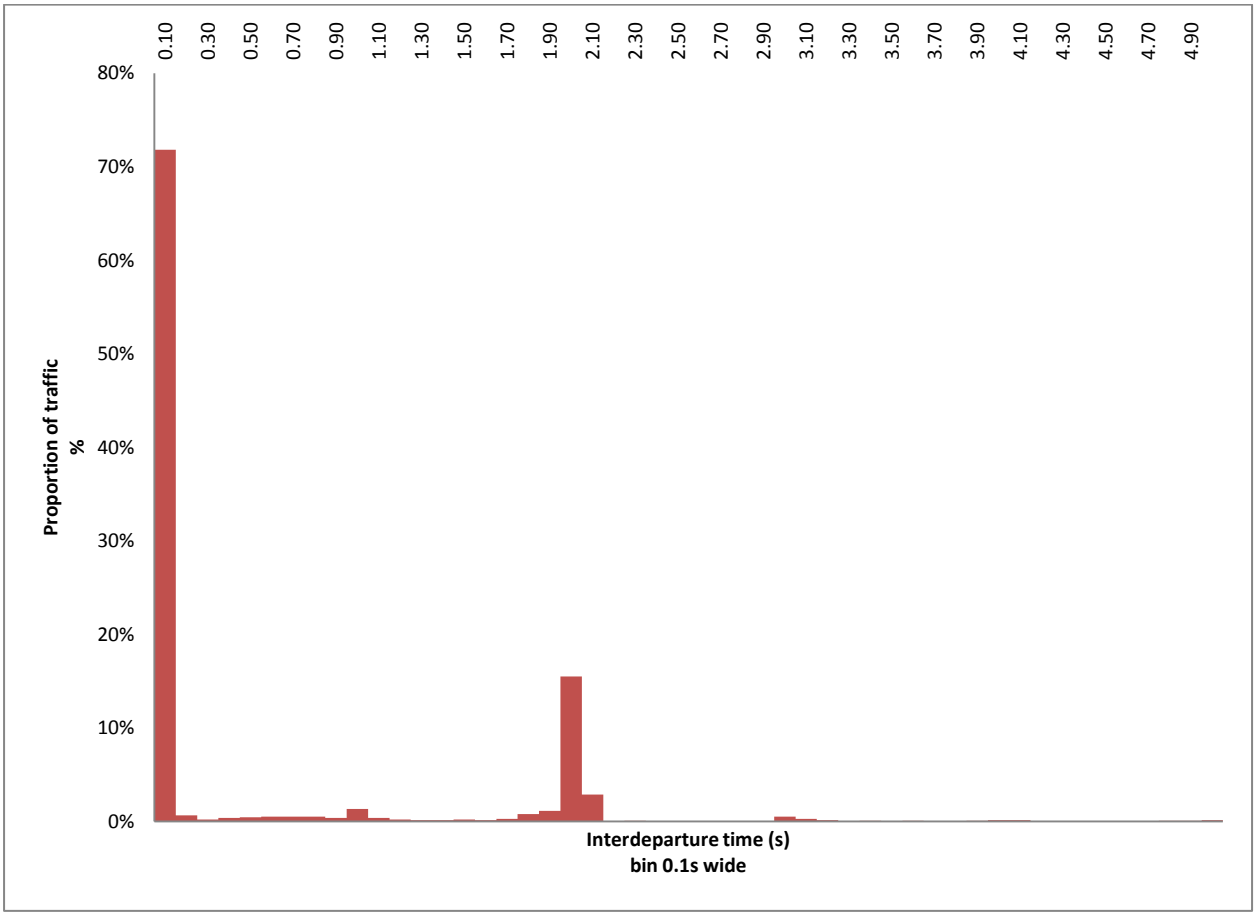

Figure 5-3: Average Interdeparture time for the MTU by RTU

We can clearly see two groupings, one under 100 milliseconds and another at around 2 seconds. This is a consequence of the polling interval. The packets around 0.01 seconds are the ACK and confirmation packets sent to acknowledge RTU communications and, as such, are heavily correlated with the sending of a polling packet. The other packets arriving between polling sequences are the result of our limited human activity (i.e. sending commands to change the values) or from delay in the server side. This abnormal or human activity is several orders of magnitude smaller than the automated network traffic and seems at first glance to be more evenly distributed across all interarrival time values. This would mean that, even on a production network the periodic components are likely to significantly outweigh human activity.

If we choose not to split the packets sent per RTU stream, we still get a heavily periodic interarrival time distribution. Figure 5-4 presents the interdeparture time of packets at the MTU.

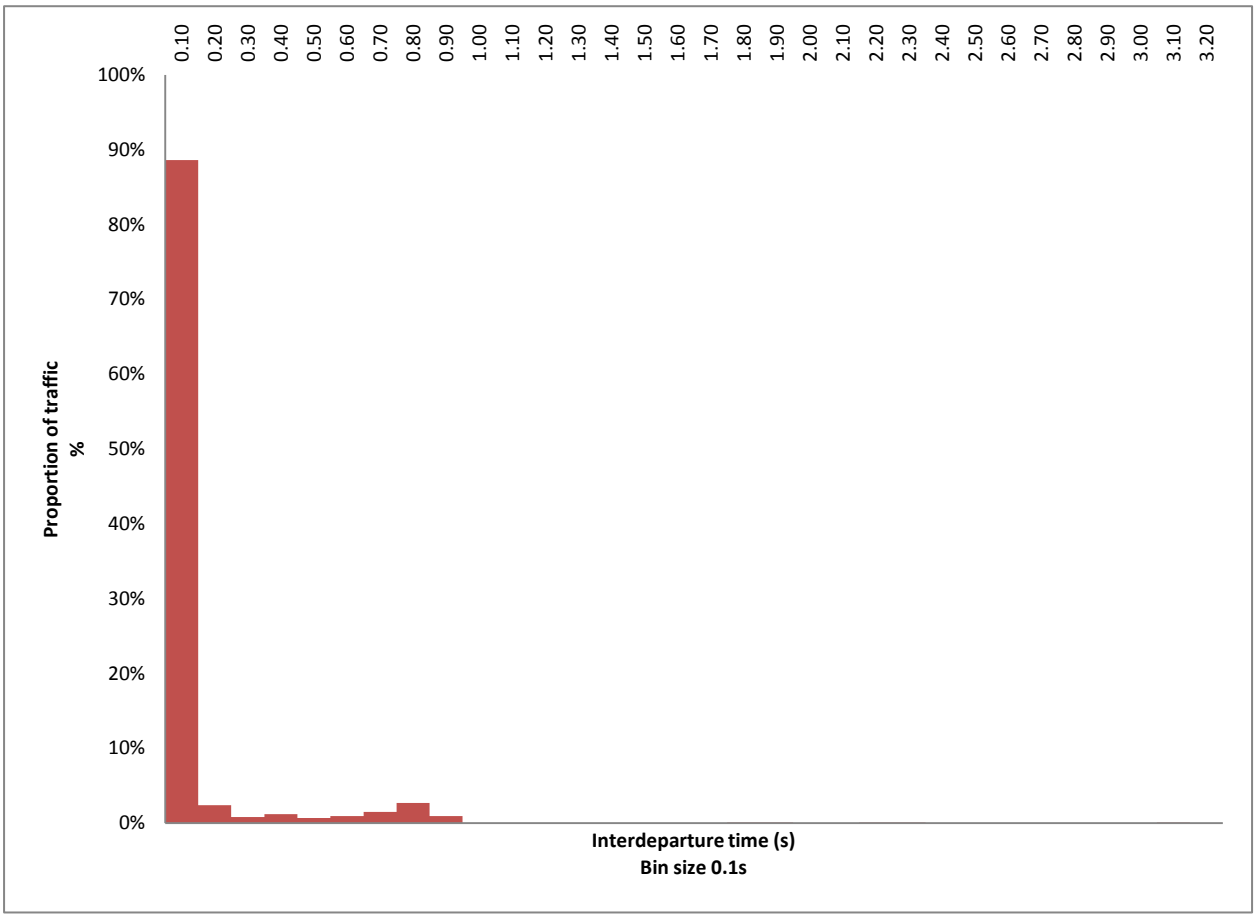

Figure 5-4: Interdeparture time MTU - multiplexed

This is similar to the demultiplexed interarrival time, but with the 2s periodic component missing. This can be explained by the serialized nature of the polling: at each polling step, the MTU sends

a polling request to one piece of equipment, going down the list. So, each 100 ms or so, the MTU receives a response from each RTU in turn creating the periodic components we can observe in the traffic from the RTU machines and in the deaggregated MTU graphs.

This is also true is we look at the traffic sent by the RTU. If we look at the interdeparture time of traffic going to the MTU for each RTU separately, we obtain a graph very similar to the graph in Figure 5-4 showing the traffic from the MTU going to each RTU. Figure 5-5 illustrates this situation. Because no packets originate from the RTU, this similarity between packets sent by the RTU and the packets sent to that particular RTU by the MTU is expected. We can still see the two heavily periodic components at around 100 milliseconds and around 1.8 seconds.

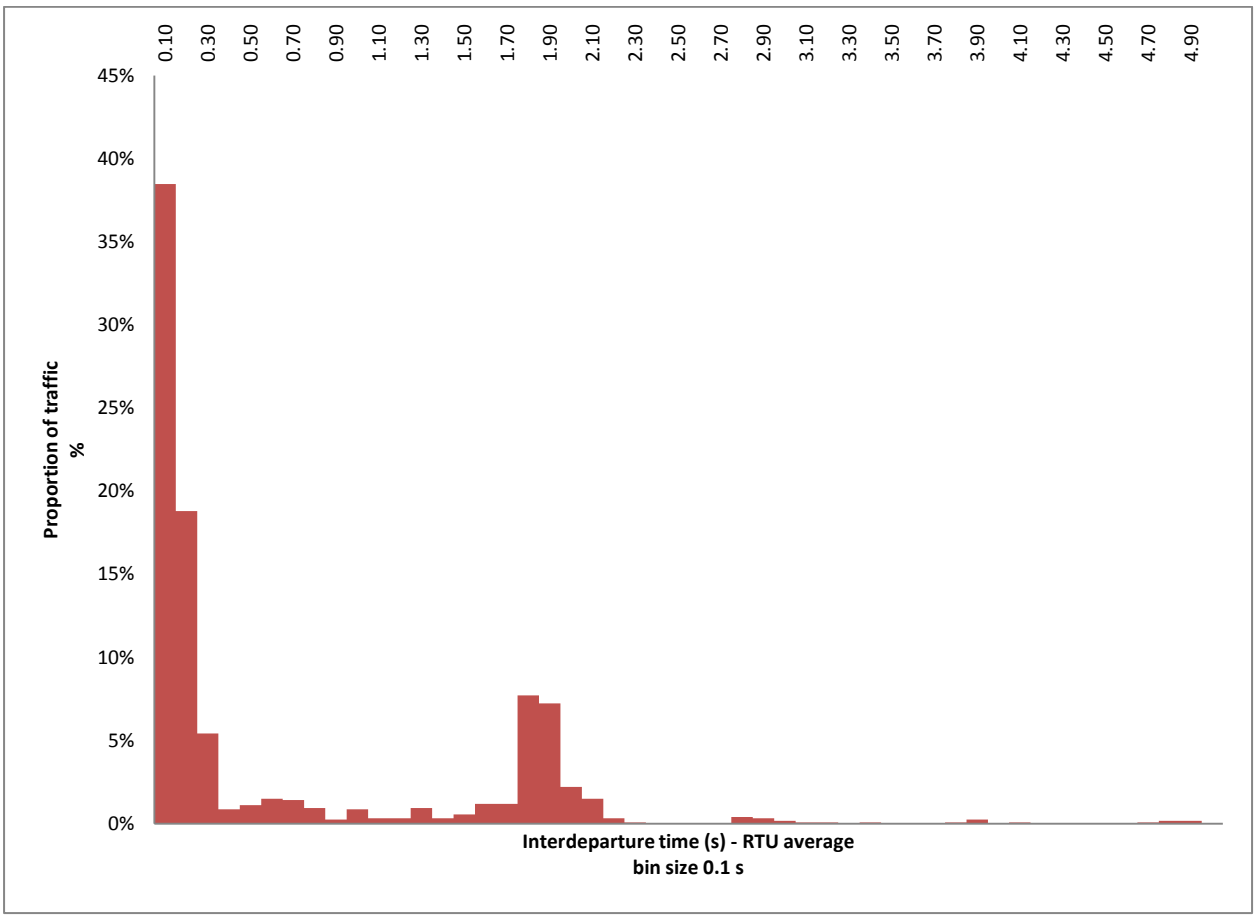

Figure 5-5: Interdeparture time RTU

5.2.4 Packet size

The use of polling also has an observable effect on the distribution of packet lengths. Because the polling requests are generally serialized, the MTU typically sends the same request to all RTUs.

In the same vein, unless there is a significant change in the operating environment, the responses will be very similar. Even if we do not perform deep packet inspection and decode the protocol, we can look at packet length to impose constraints on the traffic to create signatures. Figure 5-6 and 5-7 presents the averaged distribution of packet lengths sent by the MTU and the RTU respectively.

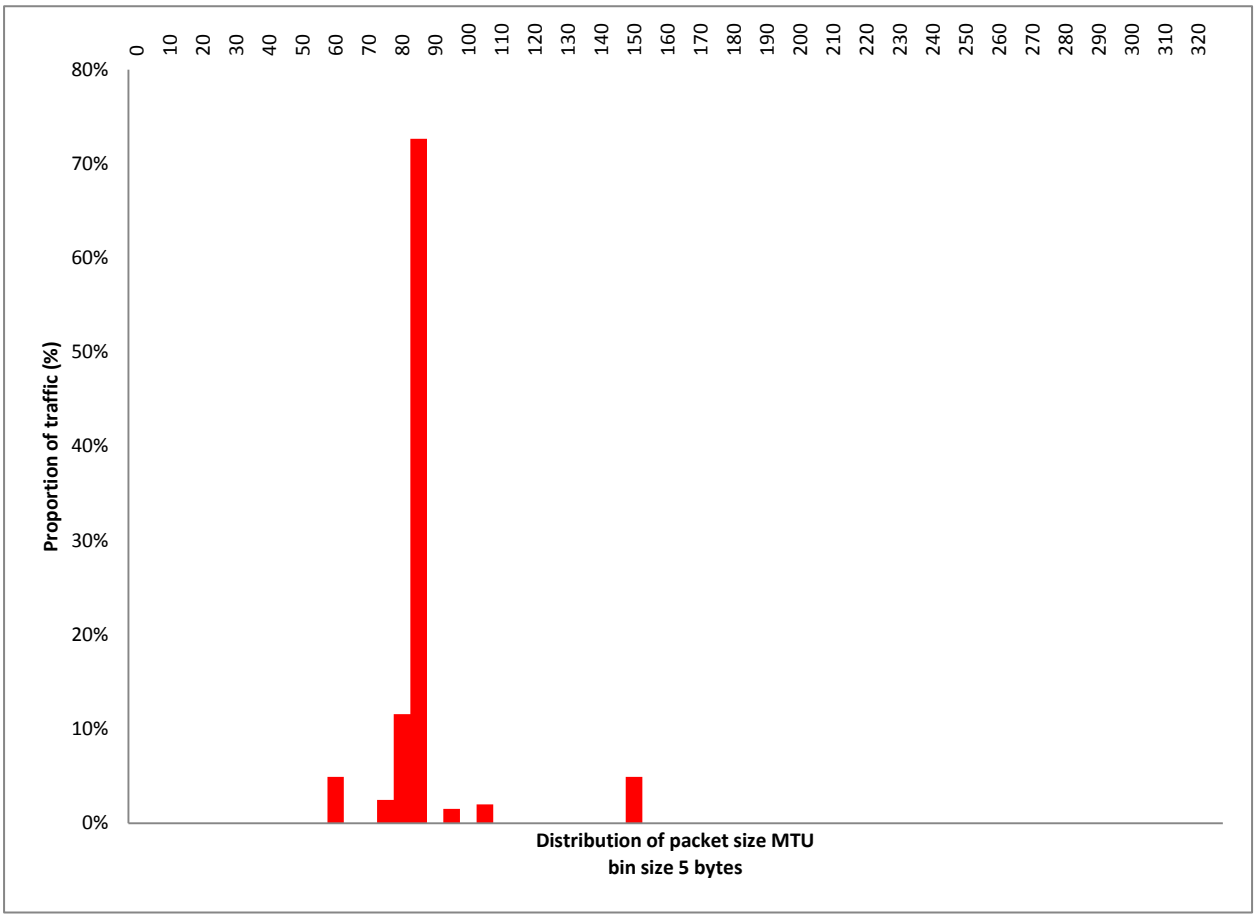

Figure 5-6: Distribution of packet lengths - MTU

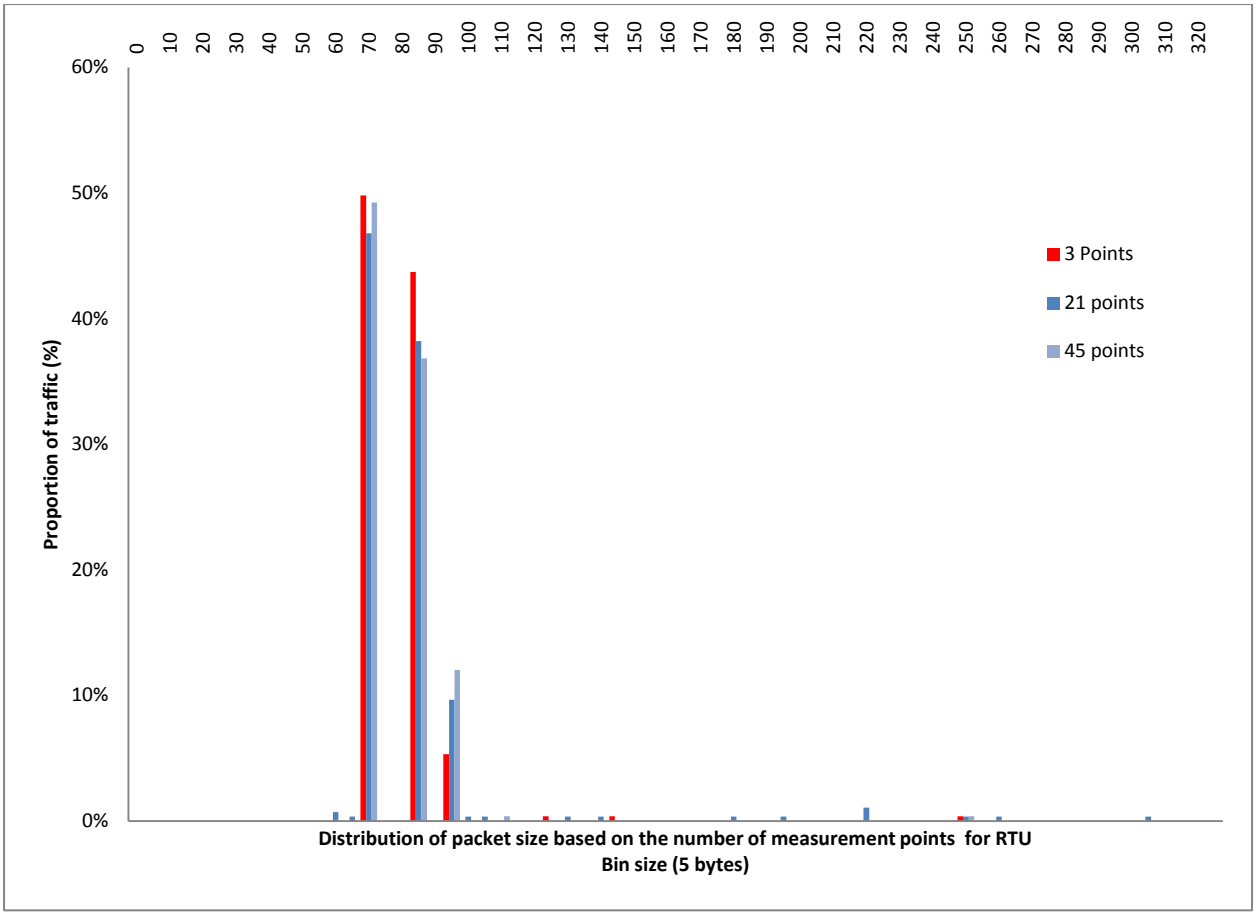

Figure 5-7: Distribution of packet lengths - RTU

We can see the vast majority of packets are of only two sizes: the size of the request or response and the size of the ACK packet used to acknowledge the request or response. The is also a small number of packets with larger sizes to account for large requests (ex. update for multiple values at the same time) and the occasional human command sent. Naturally, a large number of measurements elicits a higher percentage of oversize responses. Even these packets are of fixed sizes based on the number of records sent and are observed a number of times even if their frequency is much lower.

The MTU's behaviour is even more regular than the RTUs. In essence, because the MTU is always asking the same questions (what is the value of point X? can you operate point Y?), it is always sending the same packet. In addition, because the order in which the polling questions are sent is deterministic, it will always ask the polling questions in the same order. Only a small ammount of variability is introduced by human operators activating control points, but the volume of those will always be marginal compared to the large number of automated requests

sent. Because of this, it is easier to see the attacker's effect on the RTUs where there is a smaller amount of traffic so small abnormalities stand out even more.

5.2.5 Sensitivity analysis

In terms of distribution, if we look at the effect of our design choices, we can analyze the sensitivity of our results to design choices. To gauge the effect, we will compare the distribution of interdeparture time and packet size of different configurations.

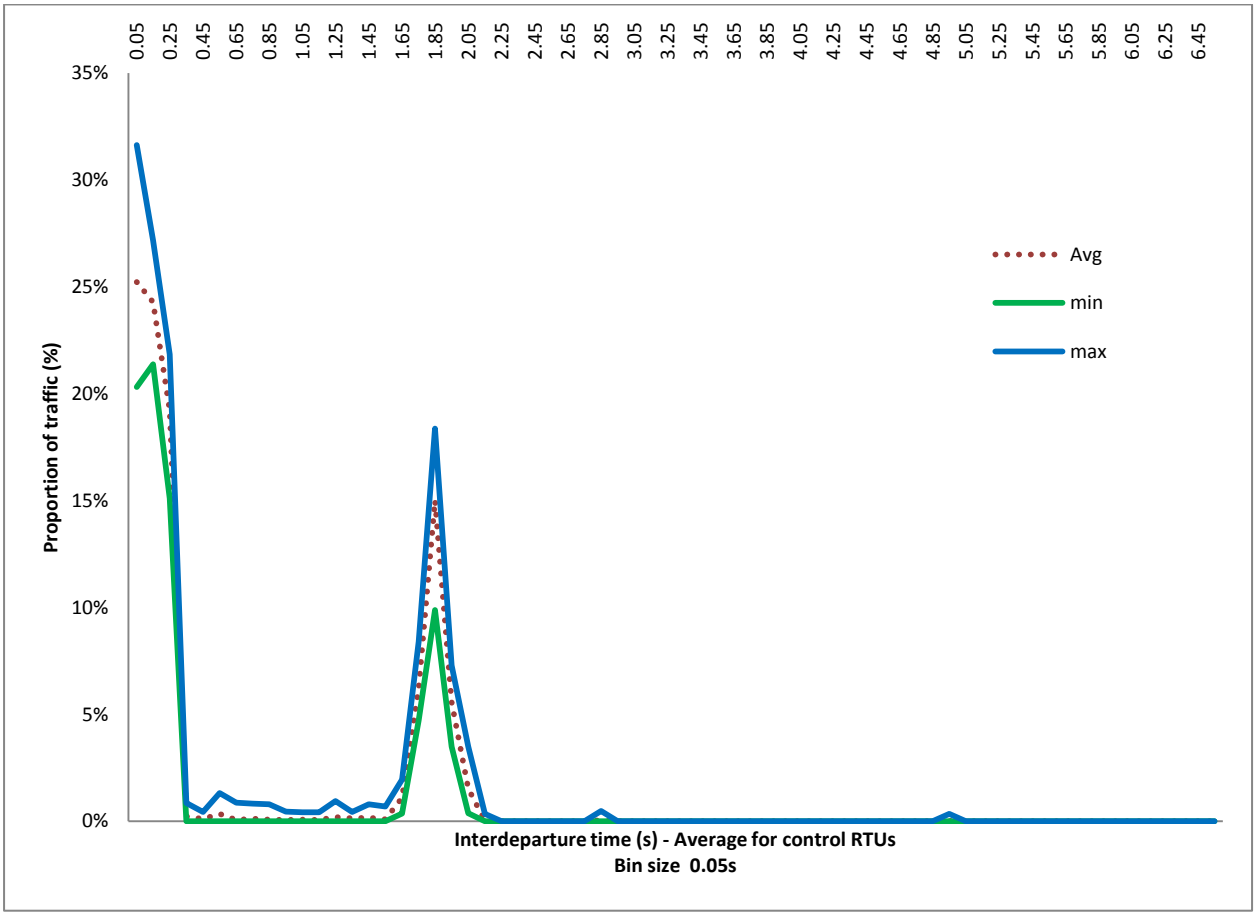

Figure 5-8: Distribution of interdeparture time for control RTUs

Our first design choice is the number of measurement points per RTU. We perform a number of experiments where one RTU has a different number of points configured in its TelDB database. We then calculated the distributions for all the experiments for one of the RTUs. In order to get a better picture of the level of variability, we calculated, over all the experiments, the maximum value, the minimum value and the average value for all the bins. Figure 5-8 and 5-9 show us the

distributions for the minimum, maximum and average values for both interdeparture time and packet size.

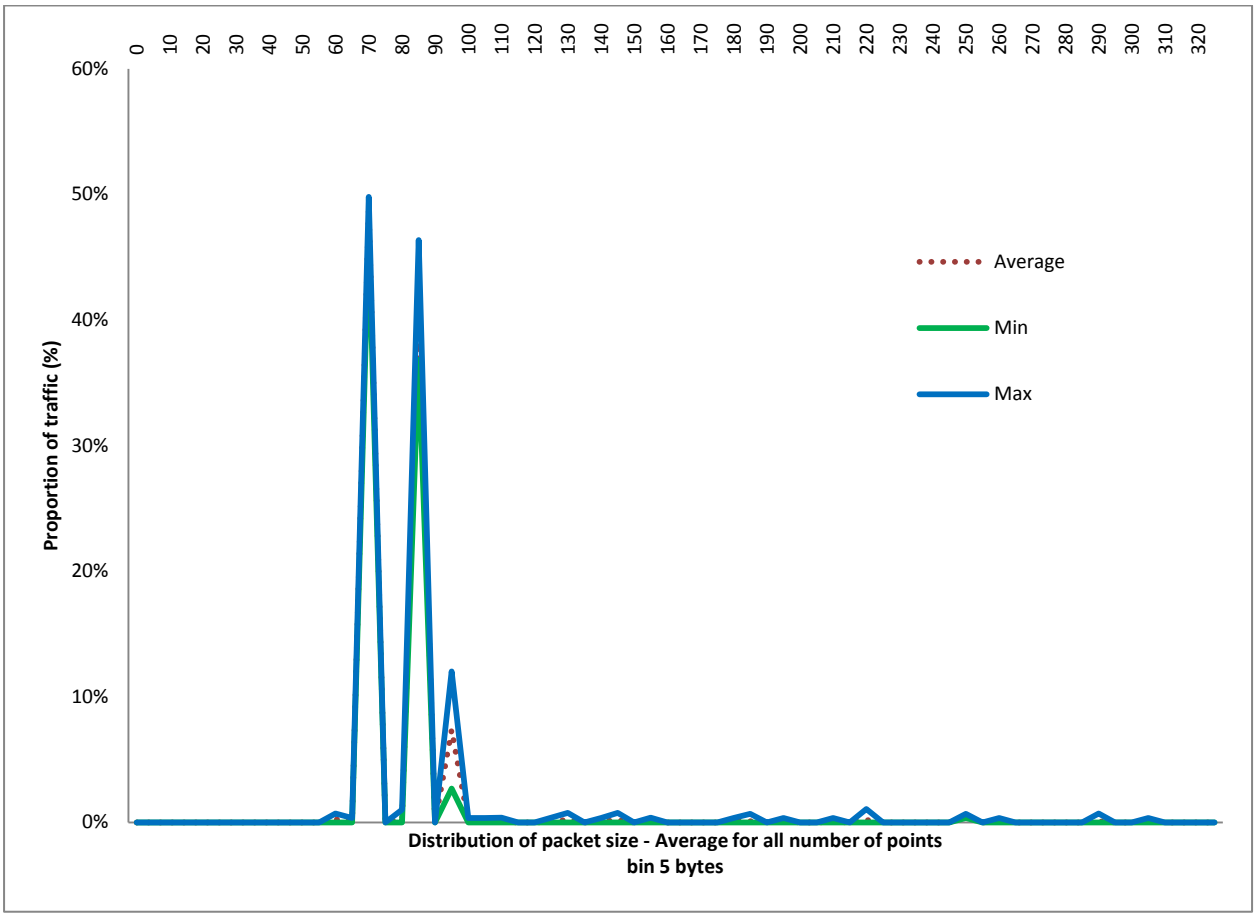

Figure 5-9: Distribution of packet sizes for control RTUs

As we can see, there is limited divergence between the traffic of similarly configured RTUs across the various experiments. We will pick one of the 8 point RTUs to act as the control distribution in a Kolmogorov-Smirnov test to determine if the number of measurement points alters the distribution of the RTU in a statistically significant way. Table 5-2 summarizes the results obtained from an online K-S calculator [130].

From these results, we can see that we require a significant increase in the number of points in order to start seeing a distance sufficiently high for us to be able to reject the NULL hypothesis that the distribution is different from a baseline of 8 points per RTU. Further analyzing the results, we find a likely cause for the sharp decline at bigger packet lengths: some packets become too big for the protocol and require the sending of additional packets.

Table 5-2: Sensitivity analysis - number of points

Metric	3 points	6 points	9 points	15 points	21 points	30 points	45 points
Interdeparture	*D*: 0.0494 *P*: 0.897	*D*: 0.0544 *P*: 0.816	*D*: 0.0439 *P*: 0.958	*D*: 0.0729 *P*: 0.433	*D*: 0.0871 *P*: 0.242	*D*: 0.0835 *P*: 0.286	*D*: 0.1088 *P*: 0.081
Packet size	*D*: 0.0269 *P*: 1.00	*D*: 0.0228 *P*: 1.00	*D*: 0.0262 *P*: 1.00	*D*: 0.0771 *P*: 0.363	*D*: 0.1015 *P*: 0.113	*D*: 0.0797 *P*: 0.339	*D*: 0.4623 *P*: 0.00

Even in systems where there is a greater variety in terms of number of PLC supported, we would see that the graph would have the same general shape, but with greater diversity for "big packets" on RTUs only. After all, all the MTU will send the same polling requests and the same response acknowledgement packets no matter the number of PLCs. For the RTUs, the proportion of acknowledgement packets and response packets will the same. The size of acknowledgement packets will stay the same, as well as is the size of responses reporting no changes. So, only the size of response packets that include records will vary. Even then, the packet size will take discrete values based on the number of records included multiplied by the fixed value of a record, up to the maximum packet size where DNP3.0 will split the packet. This will only serve to spread the tail end of the distribution over these discrete values if the sample has a large variation of packet sizes.

Table 5-3: Sensitivity analysis - number of RTUs

Metric	2 RTUs	4 RTUs	8 RTUs	10 RTUs	12 RTUs	14 RTUs	15 RTUs
Interdeparture	*D*: 0.1004 *P*: 0.252	*D*: 0.0803 *P*: 0.527	*D*: 0.0688 *P*: 0.719	*D*: 0.0573 *P*: 0.890	*D*: 0.1023 *P*: 0.213	*D*: 0.1183 *P*: 0.082	*D*: 0.0785 *P*: 0.525
Packet size	*D*: 0.0714 *P*: 0.672	*D*: 0.0712 *P*: 0.680	*D*: 0.0713 *P*: 0.677	*D*: 0.0713 *P*: 0.675	*D*: 0.0291 *P*: 1.00	*D*: 0.0558 *P*: 0.870	*D*: 0.0339 *P*: 1.00

In terms of number of RTUs, a more limited sensitivity study was done prior to experiment design. Using default configuration RTUs with a standard TelDB database with 7 measurement points, we tested the impact of the number of RTUs in the system. We compared the distributions

for multiple numbers of RTUs to a baseline of 6 RTUs to verify that there was no significant modification of metrics for a given RTU which was present in all experiments. Table 5-3 summarizes the results obtained from an online K-S calculator [130].

After further study, we determined that the serialization of communication, the fact that the MTU communicates with each of the RTUs in turn, prevents the observation of big differences in the distributions until the MTU itself comes into contention. When it does, the architecture requires the use of additional Front End Processors to eliminate the contention.

In summary, the behaviour of SCADA network equipment is driven by the master-slave architecture. In that architecture, the slaves, in our case the RTUs, can only communicate to the master, i.e. the MTU, and never communicate with each other or with other endpoints. This creates a communications table similar to the one illustrated in Table 5-1. In addition, the protocol strictly codifies the communication between MTU and RTUs, which causes the traffic to follow patterns which are very distinct from the patterns of traffic in a typical corporate network. These patterns can be identified by looking at the distribution of certain features such as packet sizes and interdeparture times.

Because neither the number of RTUs or the number of measurement points assigned to a RTU affect the distribution of packet sizes or interdeparture times the distributions presented in figures 5-5 and 5-6 truly represent the typical behaviour of an RTU, even if small statistical variations can be observed. To limit the effects of these variations, we will use the average distributions illustrated in figures 5-8 and 5-9 to act as our description of normal traffic for the RTUs and the maximum and minimum distributions illustrated in the same figures to act as boundaries for the statistical variance.

The combination of these features gives us a good portrait of normal traffic in a SCADA system against which it will be possible to detect abnormal behaviour.

5.3 Scenarios

Jumping off from the characterization of normal traffic and using our ICS sandbox for high risk experiments, we can create scenarios to test our defensive strategy for detection. We settled on three scenarios, with increasing levels of sophistication and this section describes each of the scenarios in turn. The first scenario represents an infection from commodity malware, the

Waledac botnet. The second scenario represents a hacker performing maintenance operations such as updating files using commonly used hacking tools, in this case Metasploit. The third scenario presents a sophisticated attacker using a limited bandwidth DNP3.0 covert channel.

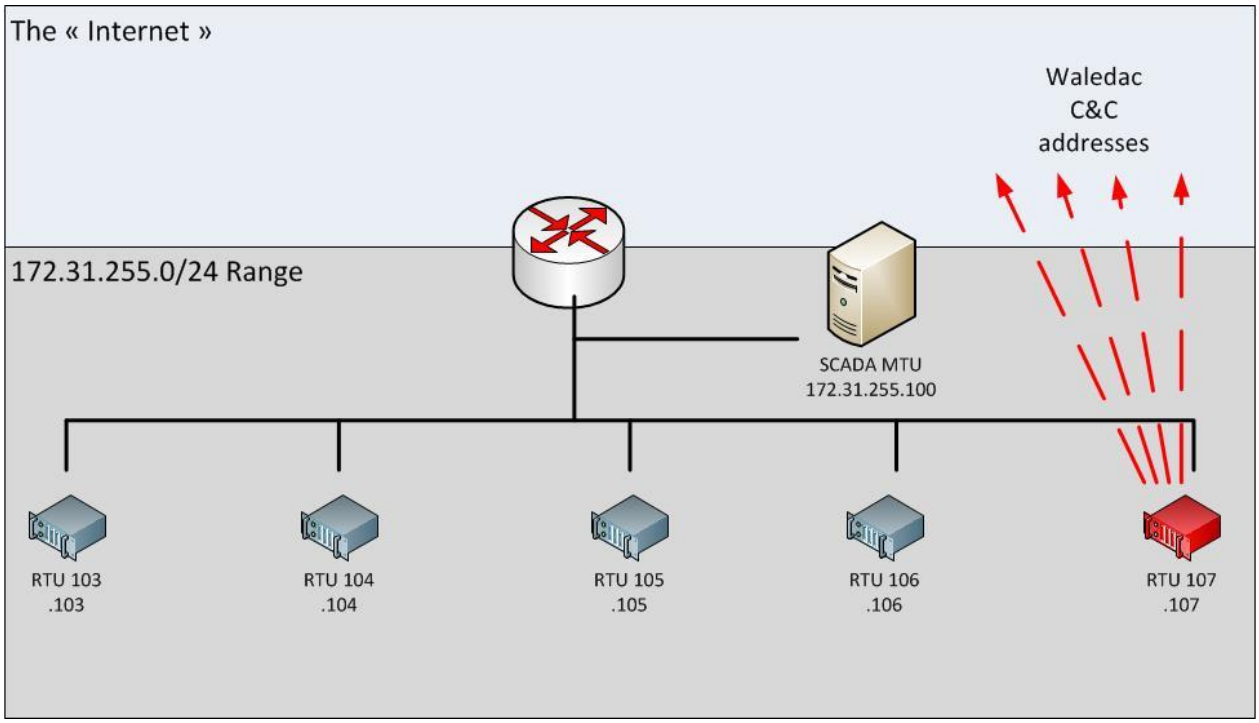

Figure 5-10: Experiment network - Scenario 1

5.3.1 Common botnet

The first scenario infects a machine with a sample of the Waledac malware [131], [132]. This particular malware was chosen because its network behaviour is well documented and because experiments were performed using a similar setup as shown in Calvet et al. [133]. Since our setup is isolated from the Internet and since we did not deploy any piece of the Waledac command and control, we do not have the full bot traffic. We have instead the beaconing traffic from the Waledac malware trying to contact a list of hardcoded IPs to establish command and control. This attack represents a very common scenario for SCADA system where a machine is infected either prior to delivery or by performing maintenance with infected equipment. The machine starts beaconing out, attempting to join a command and control server, but has no direct route. Thus, the machine remains infected for a long time. We believe this beaconing behavior is more difficult to spot in network traffic than the comparatively large volume of peer-to-peer and spamming traffic associated with an active Waledac bot. This attack scenario represents a low

level of attacker sophistication where the attacker relies on direct communication with the machine. Figure 5-10 summarizes the scenario.

5.3.2 Advanced persistent threat

The second scenario increases the level of sophistication of the attack. Instead of relying on direct contact to the Internet, the attacker goes through a compromised machine to create a pivot that enables him to access the other machines. Normally, the pivot point could be a machine that was badly configured and provided a remote access which is not directly observable (for example a through a modem or a local wireless network) or through an out of band access (such as a USB stick). The pivot node can now serve as the local distribution node for any command and control. This scenario is somewhat representative of the type of peer-to-peer C&C network that can be observed in the Stuxnet worm: a communication node with access to outside is identified and that node propagates updates to other infected nodes on the local area network. Network defenders do not see any of the telltale connections to outside addresses on abnormal ports coming from inside. However, unlike Stuxnet, we did not deploy a full peer-to-peer network. Instead, we used the pivot function of the Metasploit framework [134] to create the pivot point. Typical maintenance operations (e.g. moving files back and forth, launching processes) were performed through the Metasploit interface to generate traffic. The TCP port of Metasploit was then modified post-hoc to prevent an easy identification of the traffic through the use of port number. While this does not represent actual SCADA malware behaviour, it is common practice for malware operating in corporate networks which hide themselves in the large volume of HTTP traffic. Also, in order to generate a reasonable volume of good and bad traffic, additional RTUs were added. This has little bearing on the ability to compare results with the other two scenarios as shown in section 5.2.5. Figure 5-11 summarizes the scenario.

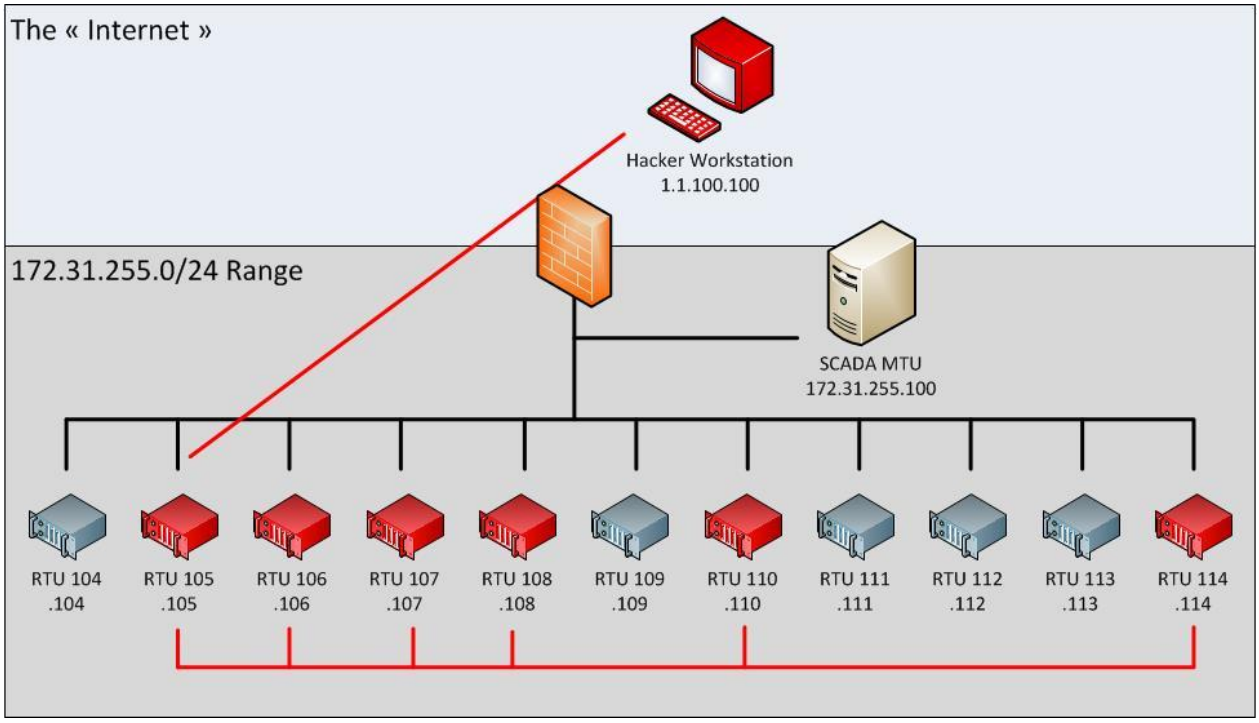

Figure 5-11: Experiment network - Scenario 2

5.3.3 Covert channel

The third scenario represents an attacker intent at hiding his presence. As in scenario 2, the attacker compromises a node to act as a pivot that allows him to push updates to other infected machines. This compromise is not observable either because of the technical properties of the backdoor (e.g. unmonitored local wireless) or by using out-of-band offline methods such as a USB key. The attacker then communicates through a channel that mimics the valid protocol used. The HTTP covert channel used by Stuxnet is a good example of the state of the art of this type of technique for web traffic. Unfortunately, there is no DNP3.0 covert channel publicly available for research. We opted to emulate this kind of behaviour by creating our own channel over DNP3.0 which would represent a malware deployed on RLS103 sending upgrade information to another malware on the MTU.

By analyzing the DNP 3.0 protocol, we observed that measurement updates contained a 16 bit field representing the new value of a point. Let's say that a reported voltage value is 24.94kV (the nominal value for a Hydro-Quebec standard medium voltage network [135]). The 16 bit representation is 0110 0001 0110 1100. If we use the 4 least significant bits of the value to send data, we have 0110 0001 0110 CCCC where C is a bit of covert data. So, we now have values

going from 24.928kV to 24.943kV. This could be interpreted by an operator to be a normal fluctuation of the values coming from a number of non-malicious sources from variations in customer demand to space weather. Another way of sending traffic is to add bogus measurement points which are not known to the MTU. Because they are not known to the MTU, they are never stored in either the MTU database, the graphical display or on the historian. However, the RTU may still elect to send that measurement as an update and automatically sends the value if the MTU requests a general update (which it does approximately every 60 seconds in our configuration). Naturally, the more bogus values used and the more bits per value transferred, the more noisy the channel is. Too many bogus values and the real values are never selected for an update. Too many bits transferred per value and the more noticeable the effect is. In order to test multiple levels of attacker stealth, we settled on 8 values and we tested multiple numbers of covert bits C. This gives us a channel bandwidth of (C/2+ 8C/60) bits per seconds assuming a polling requesting an update every 2 seconds and an update of all the points every 60 seconds which corresponds to the default values used in our SCADA setup.

In order to implement this, we modified the simple electrical simulator and the RTU update script to update the RTUsim database with values based on the hex values of a compressed executable. By changing the values in the database based on the content of the coded communication, we ensure that the packets generated by the RTU strictly adhere to protocol standards while still carrying our covert communication encoded in the measurement values. This type of channel represent a channel that is established after infection to maintain command and control and provide a path to perform routine maintenance, such as propagation a new version of malware. The return communication from the MTU was not modeled because of the technical complexity of trapping the proprietary software and the multiple configurations of acknowledgement/retransmit signals that could be implemented within the DNP3.0 protocol that would require a full protocol parsing to detect. Figure 5-12 summarizes the scenario.

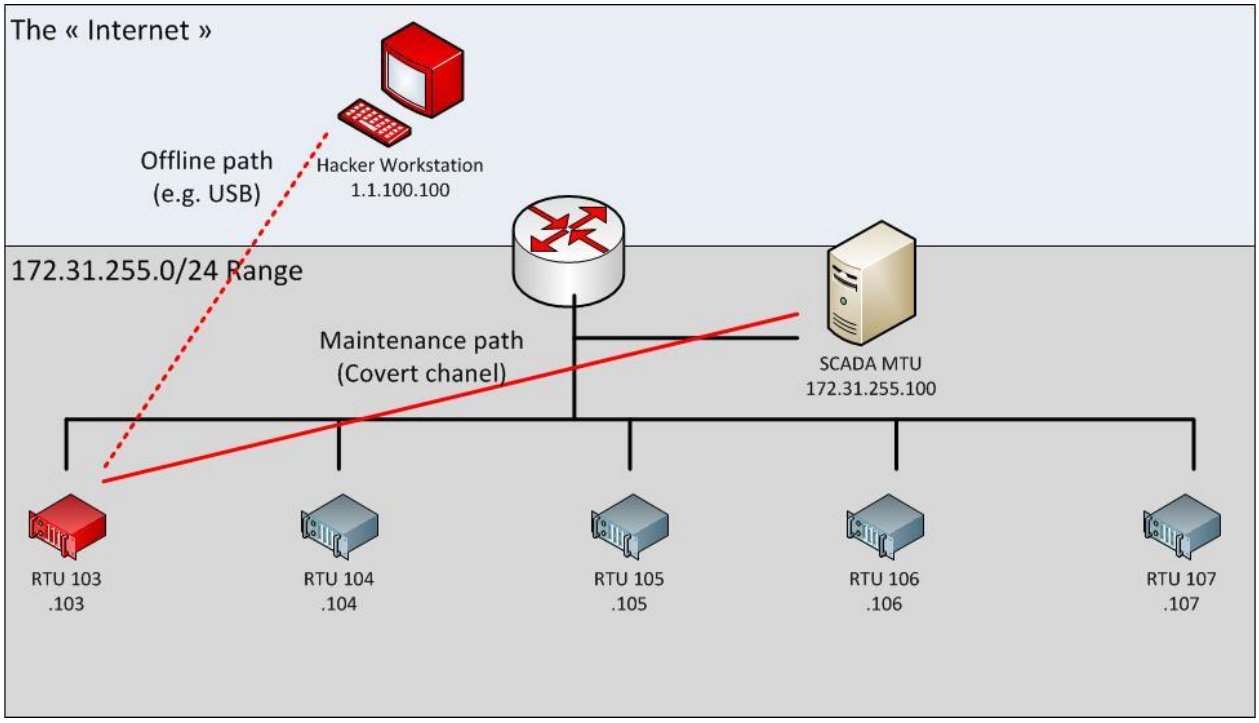

Figure 5-12: Experiment network - Scenario 3

5.4 Results

For each of the attack scenarios, the ICS sandbox was used to generate a network trace. In order to evaluate the effectiveness of an anomaly-based intrusion detection using these features, this trace was then analyzed using the selected features and compared to the baseline to see if the attackers actions stood out against the backdrop. We start by showing that the botnet in scenario 1 produces traces that are very indicative of malicious activity. Then, we see that the lateral movement of a traditional APT also stands out clearly against all three chosen features. We follow by finding the threshold of effectiveness of the suggested method for the detection of the covert channel used in scenario 3, which does not stand out against background traffic. Finally, we use a more complex feature, entropy calculation, to show the level of similarity to real traffic that needed to be achieved by the attacker to retain stealth.

5.4.1 Scenario 1 - Botnet

After infecting one RTU with the Waledac malware, we can observe the infected machine's behaviour and see how it diverges from the baseline we established with clean machines.

Table 5-4: Conversation analysis with infected RLS (Hardcoded Waledac C&C in red)

IP source (A)	IP destination (B)	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Duration (s)
172.31.255.100	172.31.255.103	264	21706	235	19959	130.2
172.31.255.100	172.31.255.105	273	22459	240	20832	130.0
172.31.255.100	172.31.255.106	241	19781	219	17806	130.2
172.31.255.100	172.31.255.108	261	21465	233	19845	130.3
172.31.255.100	172.31.255.104	269	22193	241	20491	130.3
172.31.255.100	172.31.255.107	249	20449	223	18216	130.2
172.31.255.100	172.31.255.255	176	18128	0	0	129.2
89.18.58.10	172.31.255.103	0	0	9	558	50.9
119.192.145.145	172.31.255.103	0	0	6	372	50.9
83.87.159.131	172.31.255.103	0	998	3	0	9.0
117.102.35.90	172.31.255.103	0	0	3	186	8.9
69.203.207.115	172.31.255.103	0	0	3	186	8.9

Logical topology

In terms of intrusion detection, this feature can be a great asset because one of the first instincts of the malware is to try to connect to its command and control network to join the botnet. The creation of white list rules for communication within the SCADA network seems feasible in most environments where human access on the machines is rare. In other cases, threshold rules or even a simple inspection of net flows, in which a volume of communication significantly different from the other branches of the tree is observed, could be a good indicator of the need for a more thorough investigation. To illustrate this, we infected the RLS 103 machine with a sample of Waledac. Once infected, the machine immediately attempts to contact a machine in the hardcoded peer list to establish command and control. Examples of this communication can be seen in the last five rows of Table 5-4. Because we did not provide internet access, only the SYN

packets can be seen. This traffic is easily identifiable in the Wireshark conversation report presented in Table 5-4.

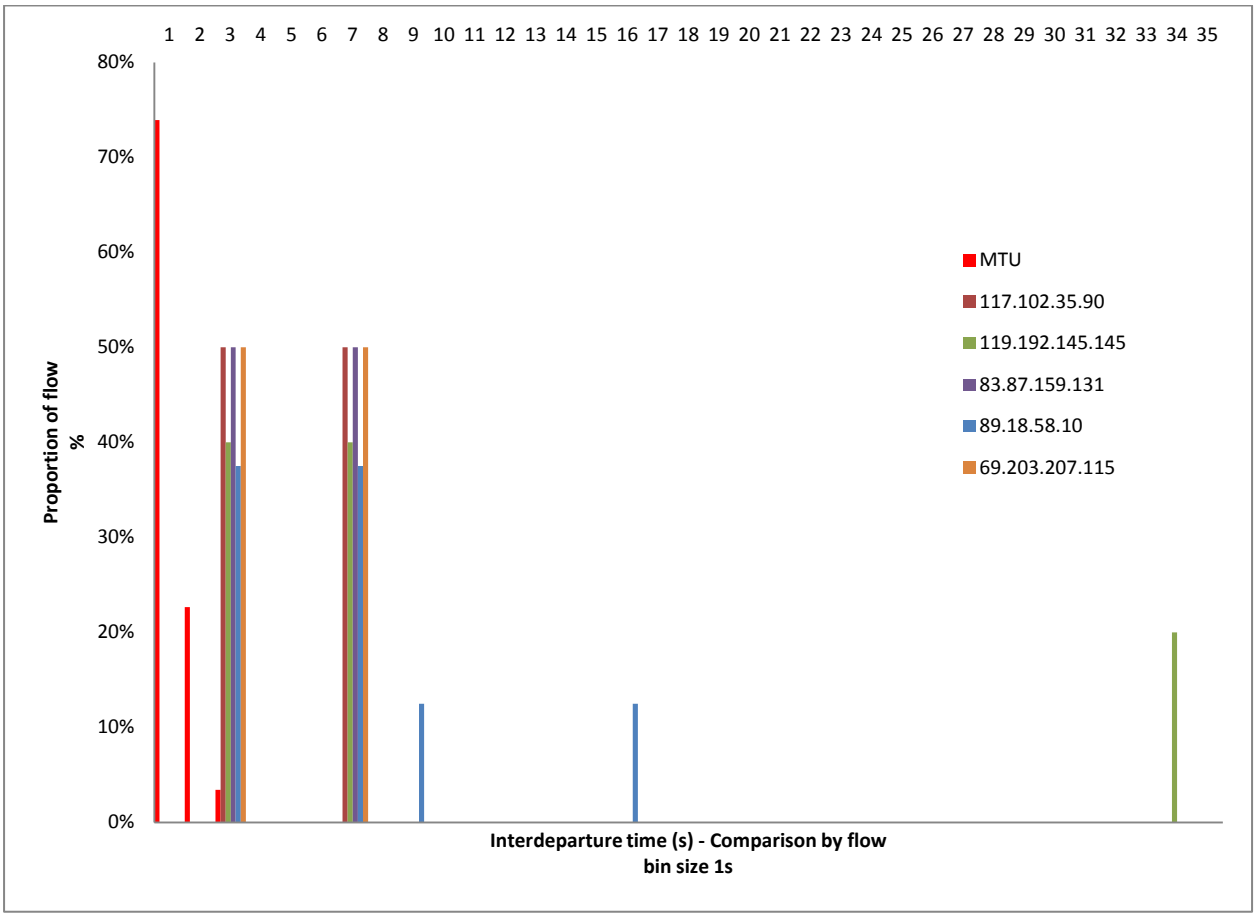

Figure 5-13: Comparison of interdeparture time for infected RLS

Polling frequency

In terms of intrusion detection, the analysis of packet interdeparture time would force attackers to synch with the existing periodic elements to stay undetected. This makes the task of an attacker manually attempting to perform post exploitation operations on a compromised machines more complex because it would create a significant volume of traffic that is located away from the two periodic components in our model of normal. Figure 5-13 illustrates the difference between the periodic components of SCADA traffic and the Waledac traffic as observed on the infected RLS. Because of the large amount of time between the C&C packets, we scaled our bins to be 1 second instead of 100 ms to increase the readability of the graph. We can see that the vast proportion of traffic going to the MTU is the periodic components we have identified. On the other hand, if we

look at the other conversations, we observe a longer time between the occurrence of packets. These features did not appear in the model of normal traffic and can be flagged as anomalous.

Packet size

In terms of intrusion detection, the small number of possible values for packet sizes would suggest that the distribution of packet length would be a good tool to detect malicious or unusual activity on a SCADA network. The possibility of observing legitimate packets associated with human operator actions which have differing lengths makes it unlikely a white list could be built without extensive protocol analysis. These types of packets are seldom encountered. It is possible that rules based on crossing a certain threshold could be built. Such rules would be able to detect tools operating continuously and sending packets of abnormal sizes. Figure 5-14 illustrates the difference in sent packet size between a clean version of the RTU and the version we infected with Waledac.

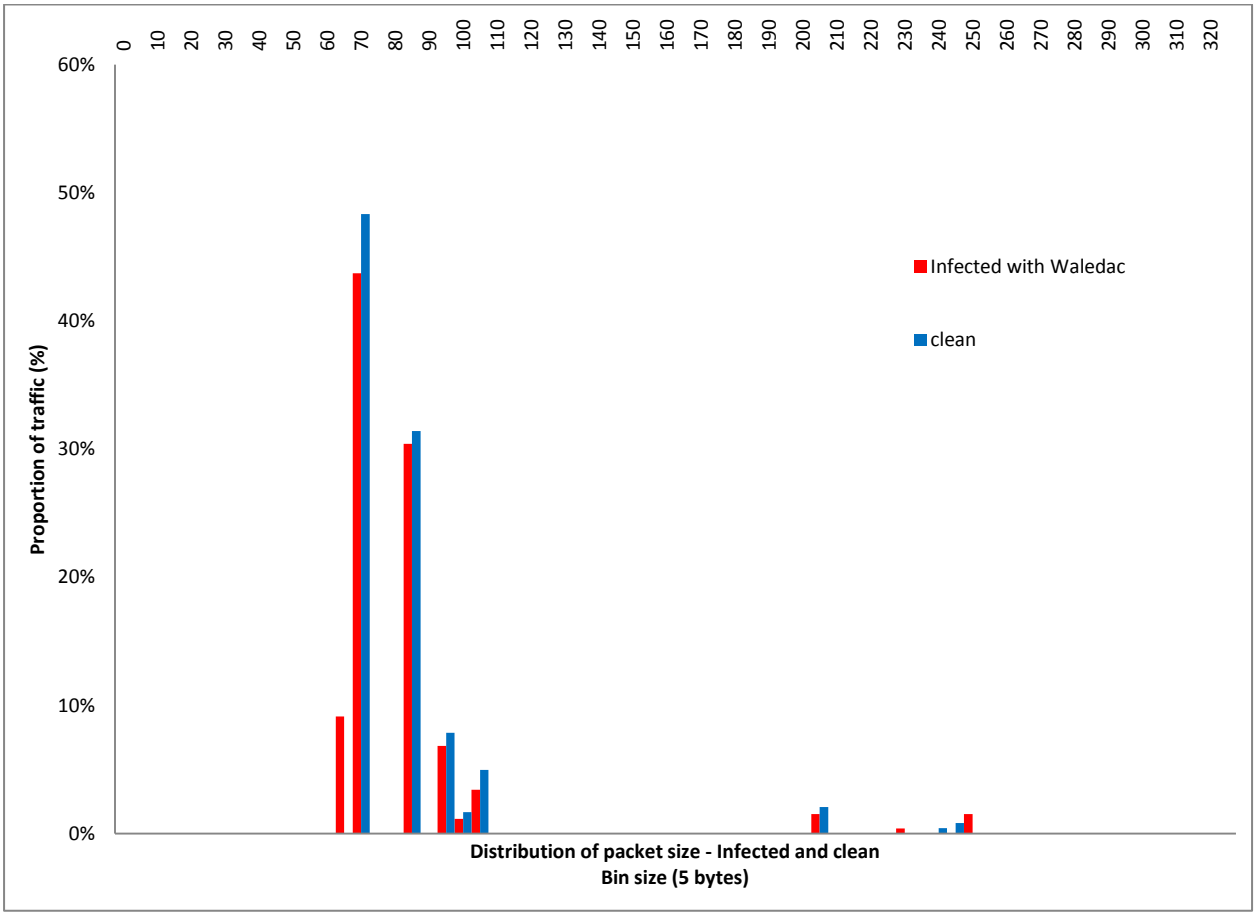

Figure 5-14: Packet size comparison between infected and clean

As we can see, the infected machine has a significant proportion of packets in the 60 to 65 bytes frame length bin (attributed to many 62 bytes packet). This is of course the size of the beaconing packet from Waledac. Because this is not a usual size for a request packet, the clean RTU has no occurrence of a packet of that size. As a matter of fact, that packet size was not observed in any of the SCADA traffic we generated for the sensitivity tests.

5.4.2 Scenario 2 -APT

After setting one RTU as a pivot point with Metasploit, we perform malware maintenance operations on other infected nodes. We can observe the infected nodes and see how their behaviour differs from the behaviour of clean machines.

Logical topology

As for the Waledac scenario, this metric is an indicator that there is something wrong. While there is no tell tale sign like machines connecting to outside IP addresses, we know from the protocol that there should be no conversation between two RTUs. The RTUs only respond to polling from the MTU. However, once an attacker gets control of a node in a sub network, he often attempts to enlarge his foothold by infecting other machines in the same network from the machine he compromised. Once these machines are infected, they often create a local command and control network amongst themselves to enable the attacker to easily access any of these machines from where he sits outside the network, usually going through the only machine he has access to, the machine initially compromised. All of this generates conversation between machines in the same sub network, in our case RTUs, which is not naturally occurring. The conversation list from scenario 2, reproduced in Table 5-5, illustrates this behaviour.

Table 5-5: Conversation analysis with pivot point (malicious conversations in red)

IP source (A)	IP Destination (B)	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Duration (s)
172.31.255.100	172.31.255.104	516	39285	479	37778	291.2
172.31.255.100	172.31.255.105	554	45464	515	40775	321.3

IP source (A)	IP Destination (B)	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Duration (s)
172.31.255.100	172.31.255.106	521	42683	482	39397	289.5
172.31.255.100	172.31.255.107	553	45375	513	40614	319.2
172.31.255.100	172.31.255.108	429	35066	396	31426	231.4
172.31.255.100	172.31.255.109	561	46039	521	41210	321.4
172.31.255.100	172.31.255.110	554	45451	513	40624	319.2
172.31.255.100	172.31.255.111	518	42433	480	38130	292
172.31.255.100	172.31.255.112	486	39796	448	35706	269.5
172.31.255.100	172.31.255.113	520	42615	483	39439	291.2
172.31.255.100	172.31.255.114	485	39630	444	35435	262.1
172.31.255.104	172.31.255.105	17	2631	16	2590	0
172.31.255.106	172.31.255.105	1133	1473475	670	192601	246
172.31.255.107	172.31.255.105	1284	1706684	667	192659	291.7
172.31.255.108	172.31.255.105	1120	1472911	607	188926	289
172.31.255.110	172.31.255.105	1130	1474539	621	189550	277.8
172.31.255.114	172.31.255.105	1121	1472767	595	188101	241.7

As we can see, the MTU establishes conversations with all the RTUs as expected. We also see all the conversations between infected RTUs and the pivot point. There is also an unexpected conversation between 104 and 105 which is actually a small Netbios exchange between two Workgroup machines to check for domain information. While this is not an attack per se, it could

be argued that it is a configuration (hardening) problem on the machines as this traffic is not required for operation. So it could be classified as grey traffic or acted upon to improve hardening on the RTUs. We can also note that the volume of this conversation is order of magnitude smaller than the volume of malicious conversations. However, this could still be used as a channel by a patient attacker.

Polling frequency

The distribution of interdeparture time for packets on the infected RTUs presents no doubt as the abnormal nature of the communications. Figure 5-15 presents the distribution of interdeparture times for all the infected RTUs and compares it to the average distribution of a clean RTU we established in our baseline.

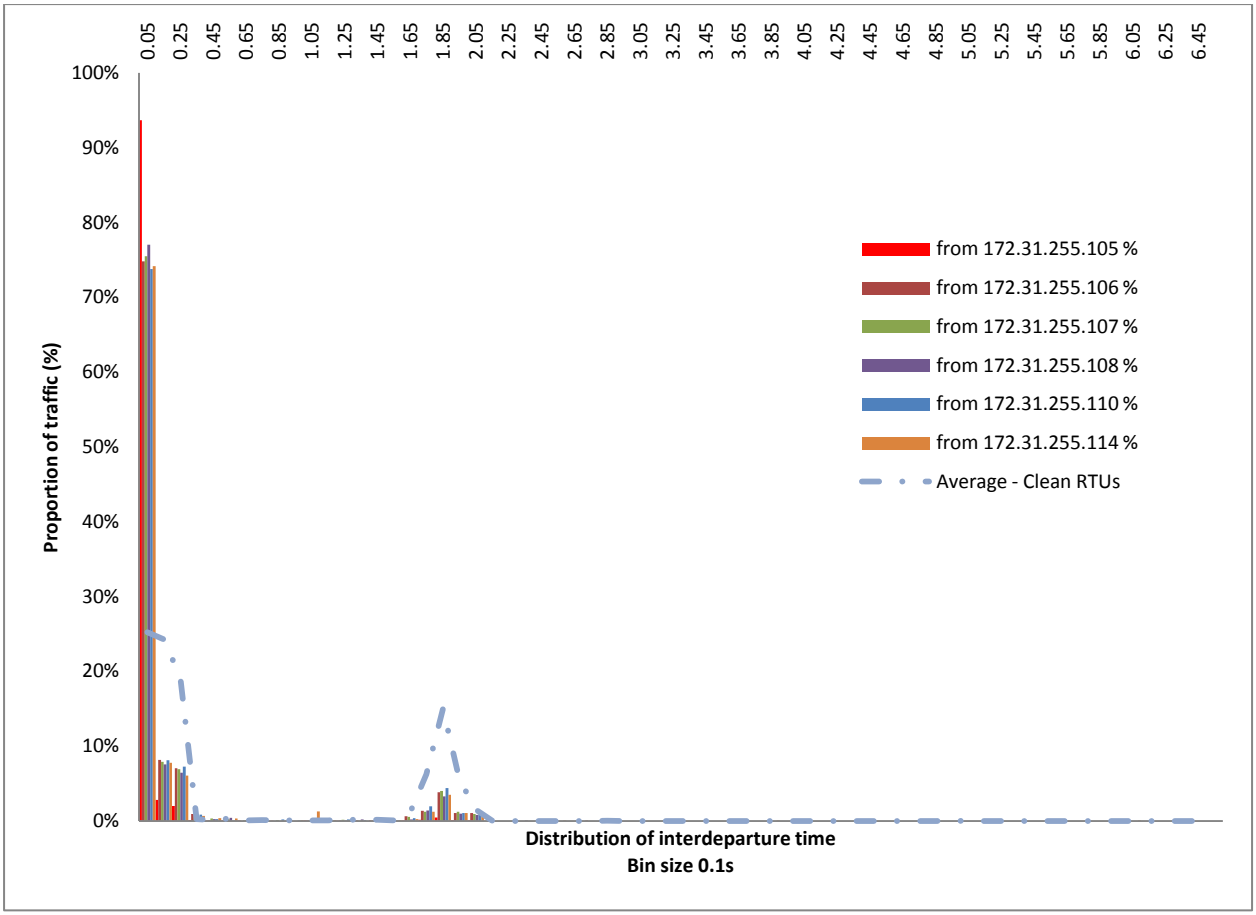

Figure 5-15: Distribution of interdeparture time for infected RTUs compared to clean distribution

Looking at the graph, we see that the infected RTUs have a much greater tendency to send packets less than 100 ms after the previous packet. This behaviour is to be expected because all

the malicious traffic has to take place between two polling sessions, reducing the average interdeparture time. In addition, the traffic sent is not regulated by the polling speed, but only the TCP flow control speed. So, to transfer large volumes of data, for example when a new executable was pushed, TCP will send packets as fast as it can to maximize bandwidth.

Packet size

As with the interdeparture time, the distribution of packet size from the infected RTUs differs drastically from the expected distribution. Figure 5-16 presents the distribution of packet sizes from infected RTUs compared to the average distribution established in the baseline.

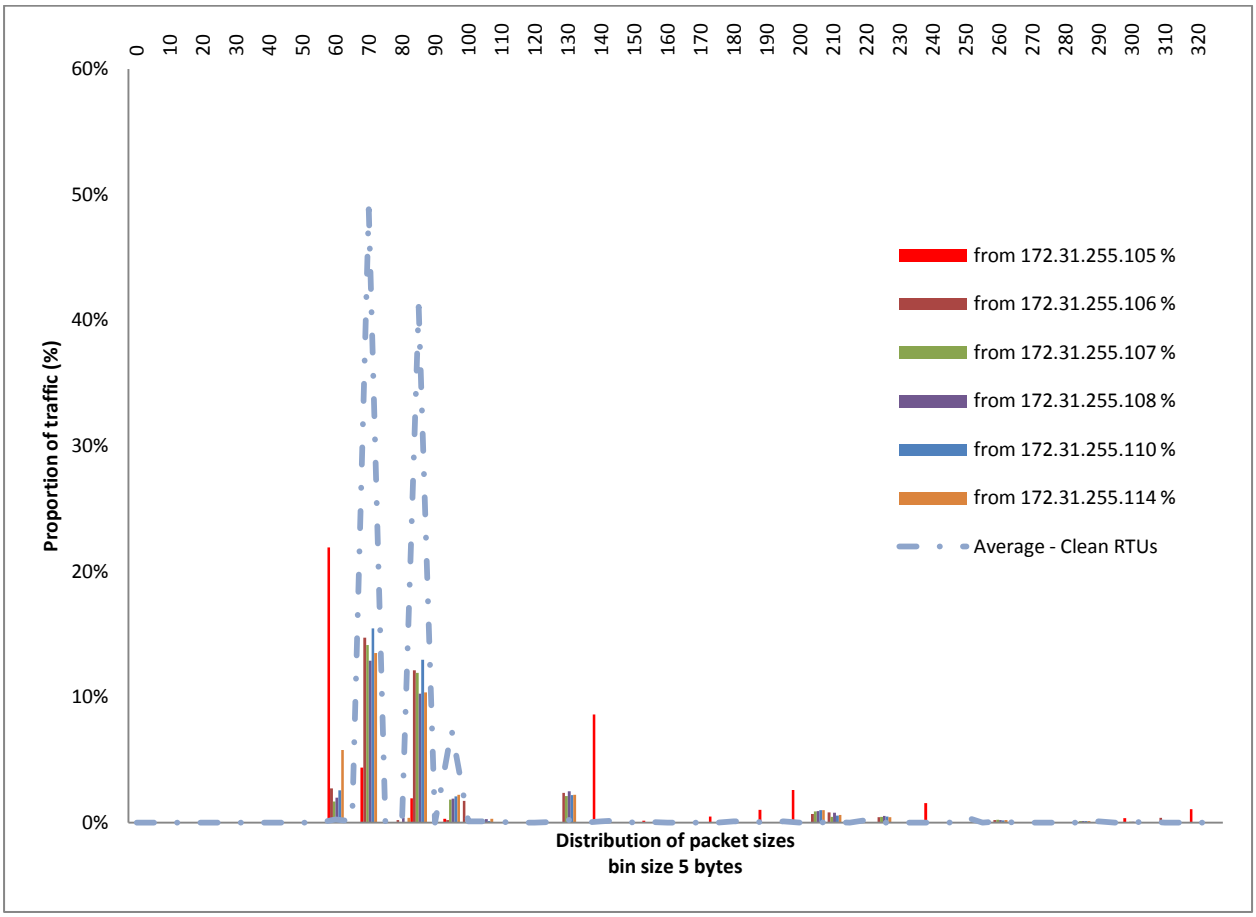

Figure 5-16: Distribution of packet size for infected RTUs compared to clean distribution

As can be expected, the TCP packets are not as strictly constrained to specific packet sizes as are the DNP3.0 packets. This spreads out the distribution of packet sizes for all infected RTUs and generates packets of sizes that are just not normally produced by the DNP3.0 protocol.

Of course, the attacker could attempt to match the statistical distributions we established in our baseline for conversation pairs, interdeparture time and packet sizes. We see one example of this in scenario 3.

5.4.3 Scenario 3 - covert channel

After installing our software that alters the measurement values to carry data on an RTU, the attacker can now use the measurement values of that RTU to transfer data without violating the constraints of the DNP3.0 protocol.

Conversation analysis

Unlike in the previous two scenarios, there is no obvious sign of malicious traffic. All the RTUs communicate only with the MTU and no extra packets are sent. This makes detecting this kind of communication difficult to spot using conversation analysis. Table 5-6 illustrates this by reproducing the conversation table from the experiment using the 9 LSBs as a covert channel.

Table 5-6: Conversation list - covert channel experiment 9 LSBs

IP source (A)	IP destination (B)	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Duration
172.31.255.100	172.31.255.103	1117	91013	937	72160	582.3
172.31.255.100	172.31.255.104	947	77807	899	67855	553.3
172.31.255.100	172.31.255.105	1009	83109	974	73363	610.5
172.31.255.100	172.31.255.106	977	80378	938	70679	581.6
172.31.255.100	172.31.255.107	703	64476	740	56196	434.3
172.31.255.103	172.31.255.104	11	1565	10	1237	0
172.31.255.105	172.31.255.104	17	2631	16	2386	0

Studying the chart, we might be tempted to look closely at 172.31.255.104 because of the grey traffic, while the compromise node is actually 172.31.255.103. The only clue as to the abnormal

nature of communications from 103 is in the slightly over average number of bytes transferred for the duration. This is cause by having a larger proportion of response packets containing updates than our normal sample.

Polling frequency

Because of the manner in which the channel is constructed, we should see no discrepancies in terms of distribution of interdeparture time. In fact, the channel is piggybacking on top of regular communications and thus uses the same timing. Figure 5-17 presents the distribution of interdeparture time for the various covert channel experiments.

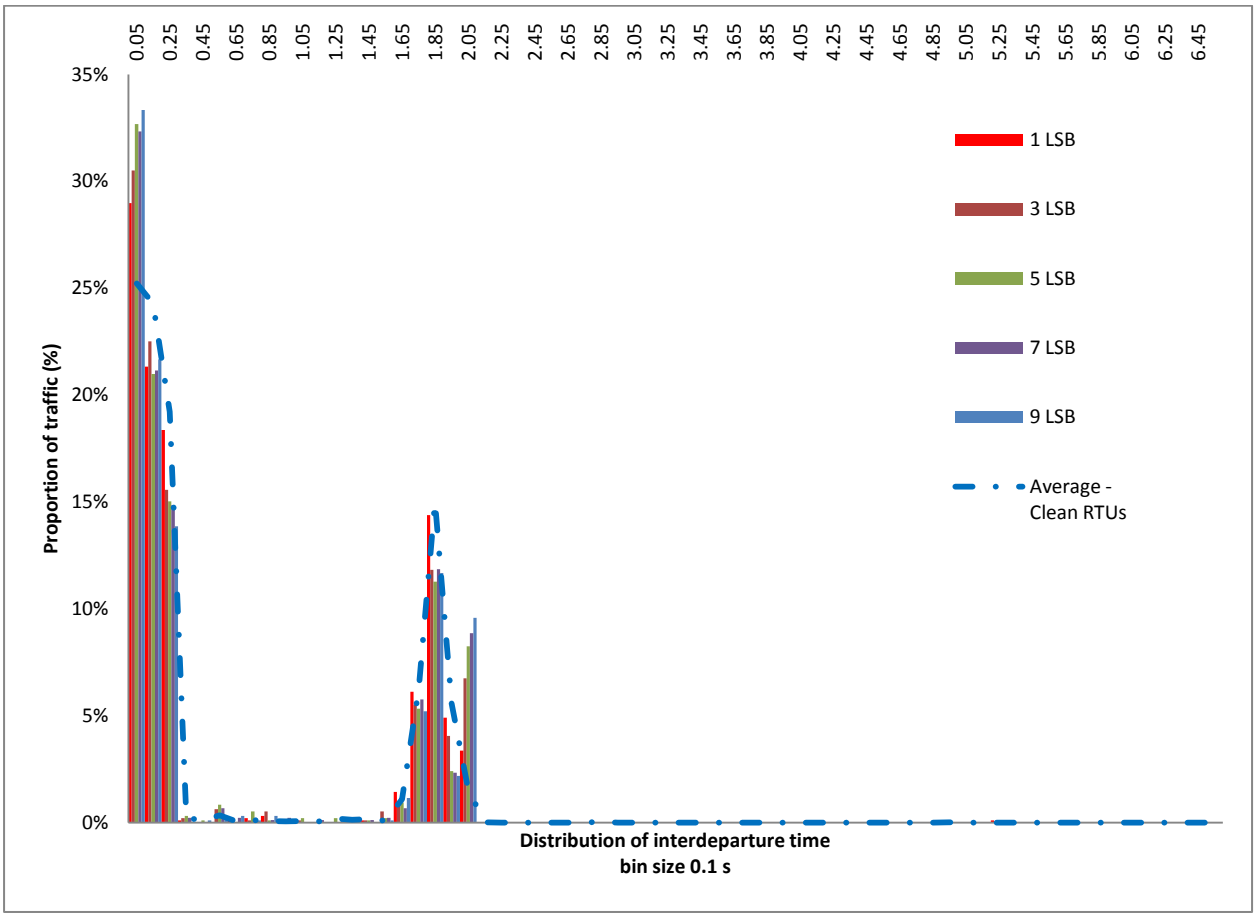

Figure 5-17: Distribution of interdeparture time for covert channel compared to clean distribution

As we can see, the distributions are not wildly dissimilar to the model as we have seen in previous cases and are within the maximum variation envelope of the baseline. As expected, this metric provides little information on the presence of a malicious channel.

Packet size

As with the distribution of interdeparture, the distribution of packet size should follow legitimate parameters since the channel is piggybacking on top of the protocol and follows protocol rules. Figure 5-18 presents the distribution of packet sizes for the various cover channel experiments.

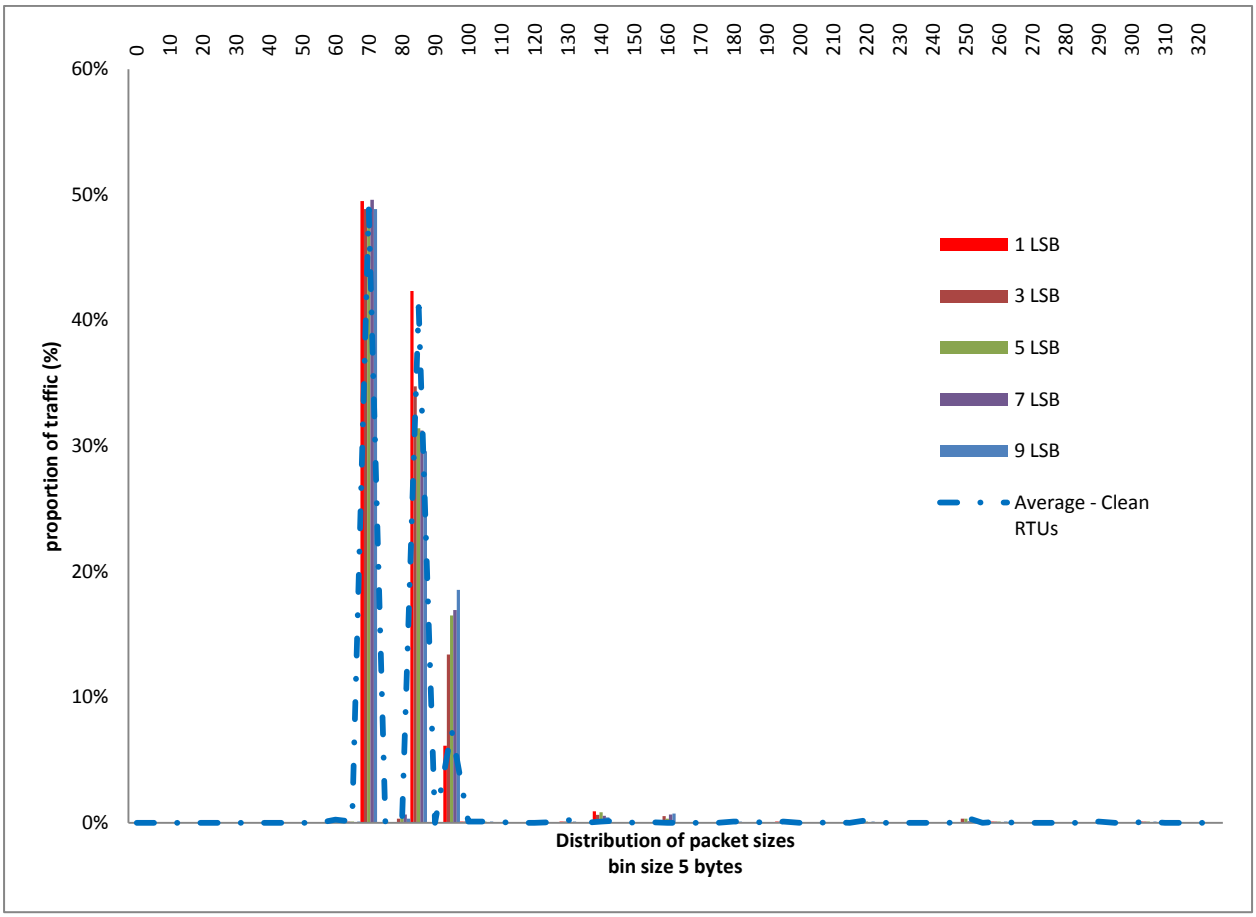

Figure 5-18: Distribution of packet size for covert channel compared to clean distribution

As we can see, the distribution of packet sizes follows more or less the standard distribution. The one exception is the greater proportion of response traffic containing data because the channel always changes the value of the measurement point. While this metric can be used to detect against our sample, we are again falling victim to the limited noise model for our baseline data. It is likely that data from a live deployment would present a level of variation of measurements that would make it very difficult to find a statistically significant increase for the channel distribution.

5.4.4 Entropy measurements for covert channel

Detecting this covert channel against regular traffic is hard. This is easily explainable by the piggybacking of the malicious traffic over regular communication. It is designed to match the normal communications very closely. Normally, even if common features would not be useful to find the malicious traffic, advanced features, such as packet entropy, might be used to identify covert channels. For example, entropy-based techniques can be used to detect very covert timing-based channels over the Internet [136]. In this case, even entropy-based techniques fail in detecting the channel showing how close the distribution of symbols from the covert channel is to the source.

In order to analyze the difficulty of detecting the channel using entropy, we need to refine the model of the source, i.e. the electrical network, to represent the correct source entropy. Our simplified simulator produces constant values, much like a steady state simulator would. However, real electrical grids cannot produce such clean power and the measurements always include some small variations caused by the vagrancies of demand, electromagnetic disturbance, solar activity and so on. In order to model the higher variation of an actual source, we modified the simulated source in order for it to have constantly changing values and create a new baseline against which to analyze our channel.

The amount of variation from the source will impact the traffic properties. To replicate the difficulty of identifying the channel on a real system, the source should have variations representative of the variations that can be measured by a PLC on a real system. While we do not have data from real systems that would enable us to build a distribution, we can find descriptions of the distributions based on measurements of high voltage lines in the literature. In their paper, Reinhard et al. [137] describe the voltage variation of the synchrophasor of a 765 KV line based on 2400 measurements. Over these measurements, they obtain a mean of 1.0003 p.u. (per unit voltage) and a variance of $7.062 \cdot 10^{-8}$ V_{pu}. Based on those values, we modified our simple electrical simulator to follow these parameters by integrating a Gaussian distribution with $\mu = 1.0003$ and $\sigma = \sqrt{7.062 \cdot 10^{-8}}$.

With the source that includes Gaussian noise, measurement vary all the time. This means that it becomes even more difficult to identify the tunnel using packet lengths. Figure 5-19 shows the new comparison with the new packet length baseline. The baseline now shows the same spike as

the channel for packets containing data. Only the channel with 1 bit differs from normal because of the small number of bits used which creates a higher number of "no change" packets.

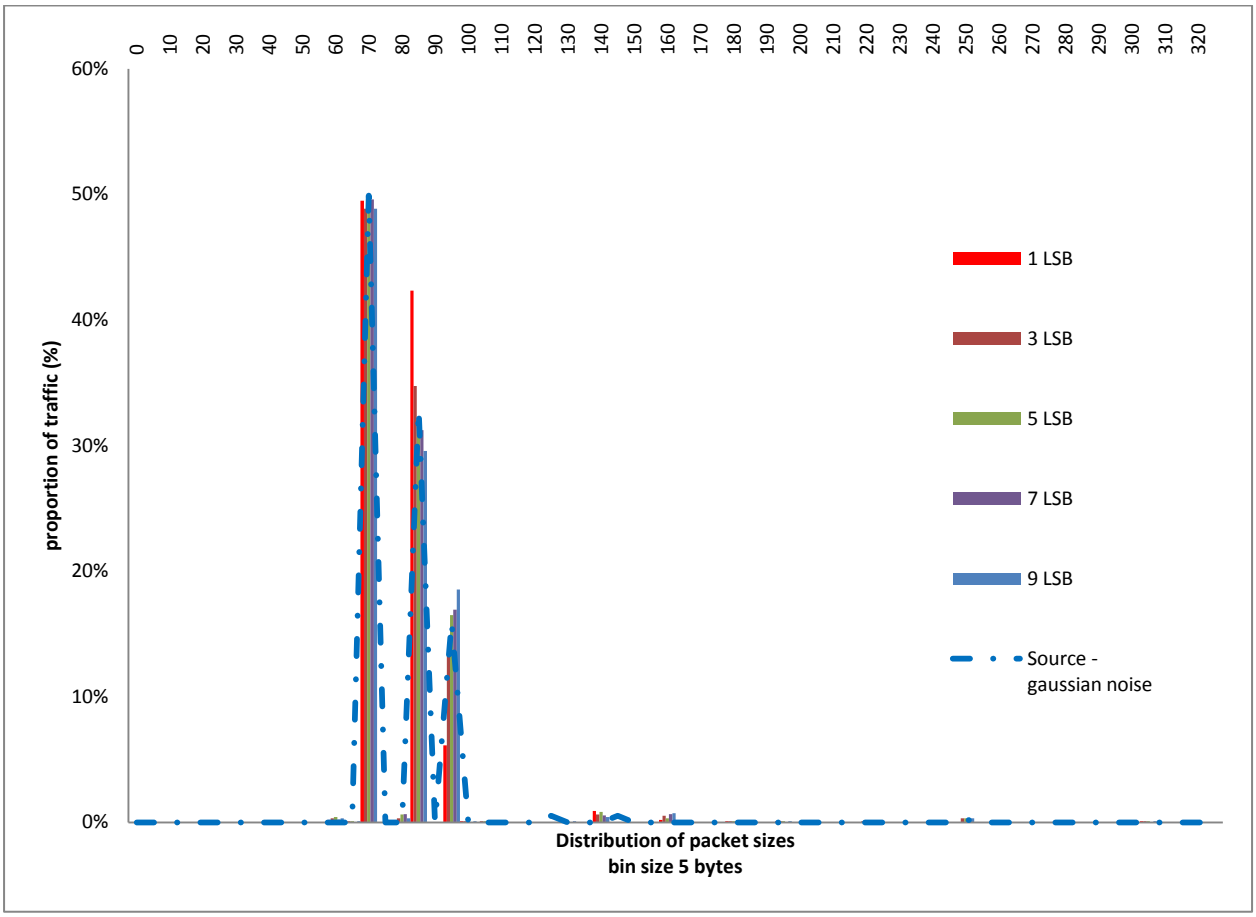

Figure 5-19 : Distribution of packet size for channel compared to a source with noise

We need to find new metrics to identify the channel. The natural choice is to use the pseudo-entropy of the packets. The pseudo-entropy is a statistical estimator of the entropy for a given communication. The formula for the calculation of the average entropy per symbol is :

$$\Psi(S_N) = \sum_i f_i \log \frac{1}{f_i}$$

where S_N is the message to be estimated and f_i is the frequency of symbol *i* in the message. Once the average entropy per symbol is calculated, we can multiply by the message size to get the message entropy. As seen in section 3.3, the packets contain two sources of entropy : entropy from the DNP3.0 signaling and entropy from the source. The pseudo-entropy from the signaling will be fairly constant across all packets, for example, the bits to request a read will always be set

in the exact same way. This means that the pseudo-entropy is a good estimator of the entropy of the source of the communication.

Shannon tells us that the amount of information we can carry is the entropy of the source. Attackers want to maximize their bandwidth. So, when an attacker is using a measurement as a medium for communication, we are expecting him to maximize the entropy of the measurement. Because the pseudo-entropy is a good estimator of the source entropy of a communication, we can use pseudo-entropy to evaluate the amount of information carried by a message. Therefore, the more information an attacker attempts to transport, the greater the entropy of the packet. We hope this will stand out against the baseline. Figure 5-20 shows the distribution of entropy for all covert channel sizes and for the source with Gaussian noise. Figure 5-21 shows a close up of the 140 to 180 range to improve visibility.

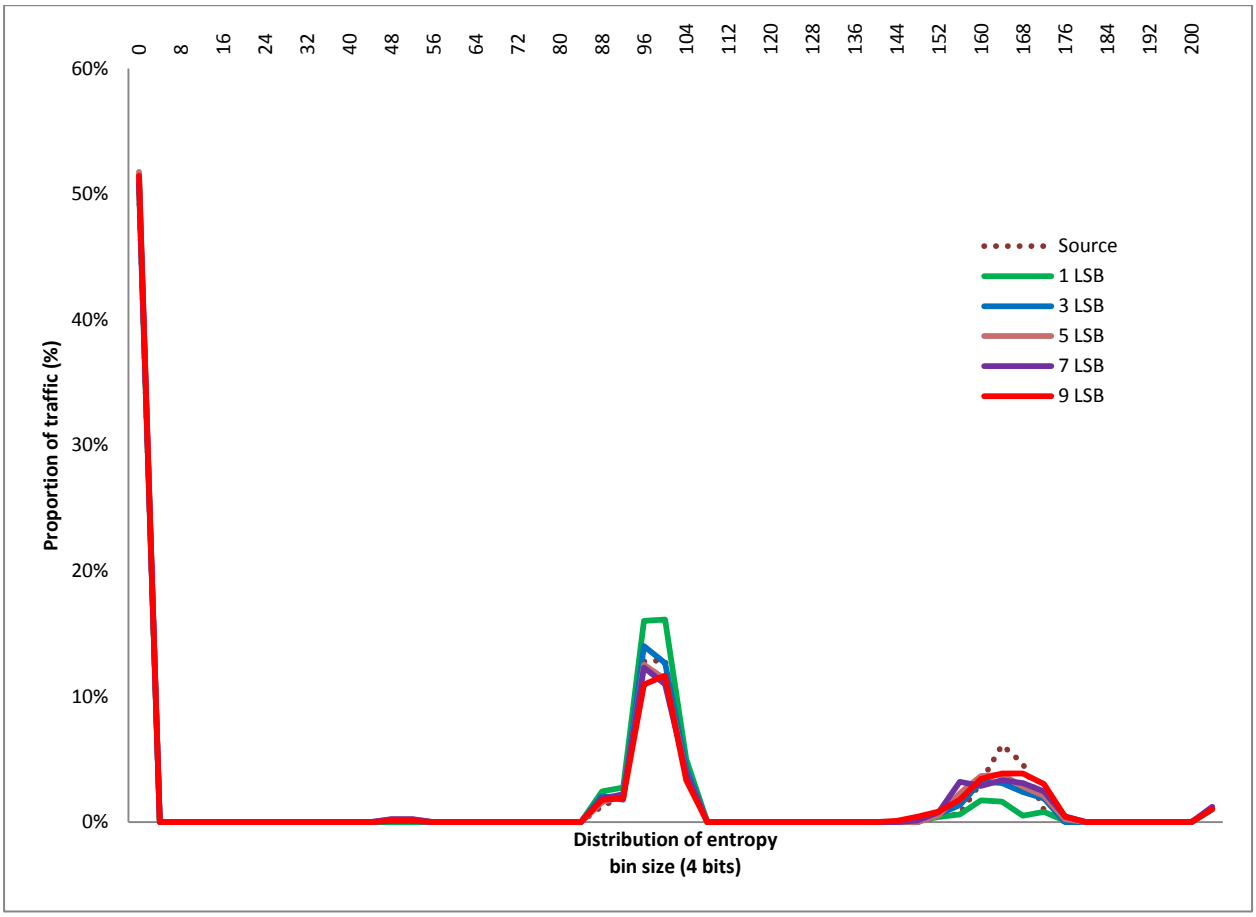

Figure 5-20: Distribution of entropy for all channel sizes and noisy source

As expected, the majority of packets carry no data, so carry no entropy. Because of the need to carry multiple OSI layers of signalling as part of the tunneling of DNP3.0 in TCP payloads, we

also observe a fairly concentrated distribution for the non empty packets. This indicates that the efficiency of the covert channel is low because we only carry a small amount of information compared to the signaling data. Because of the high amount of useless information carried, we need to zoom in to see the effect of the channel size.

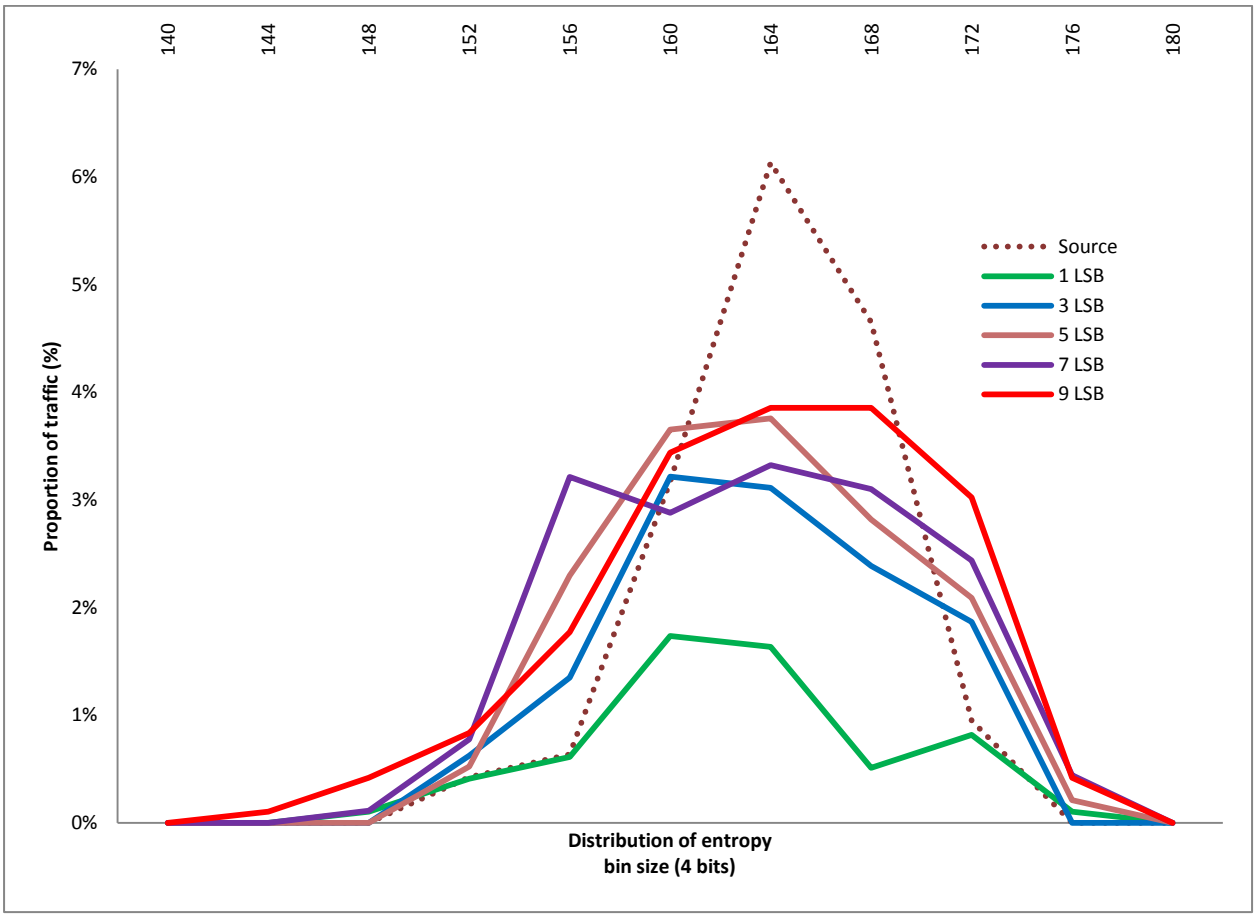

Figure 5-21: Close up distribution of entropy for all channel sizes and noisy source

As the number of bits carried in the channel increases, the entropy of the source should increase. This is illustrate in the figure as an increased weight of high entropy packets. The phenomenon can be observed in the ordering of the distribution curves in the 140-180 range. A shift in the center of mass of the curve toward higher entropy denotes a larger volume of information carried. Unfortunately, the small number of packets that are actually carrying information makes the proportion of traffic fairly small which increases the variability of the statistical estimator. The small contribution of the channel to the amount of bits transferred further complicates the problem. If all the bits of the payload would be used to transfer a compressed binary, the entropy would be near maximum for the message, meaning around 744 bits (93 bytes) for messages

containing a measurement. The studied covert channel with the highest bandwidth uses the 9 LSBs to carry data. This data comes from a compressed executable. So, in the case of our channel with the most bandwidth, only 9 bits out of 744 bits of the packet are compressed. This reduces the weight of the covert communication compared to remaining 735 bits and distorts the pseudo-entropy measurements based on the compressibility of the packet.

Comparing the distribution of entropy for the channels and comparing it to the baseline, we notice a very similar distribution. In fact, it would seem that the source conveys more information than the channels using only 1 or 3 bits. This is entirely expected because of the model used to introduce variation in the source. Using a Gaussian distribution on the measurement values is the equivalent of adding white noise to the channel. Because the standard deviation is low, this adds a small quantity of entropy. The amount of entropy injected is also relatively constant, explaining the tighter profile of the curve. In terms of values, we are affecting the least significant digits, meaning that it creates the same effect as the covert channels with a similar bandwidth. This makes it unlikely that there could be a statistically significant test that would be able to clearly identify the covert channels, especially in the face of a large variety of distribution for source noise in a production network. In turn, this means that the covert channel is nearly indistinguishable as a source from the white noise that is present in networks.

5.5 Conclusion

In this chapter, we have adapted the ICS sandbox to generate high fidelity SCADA network traffic. Because this traffic was a good representative of SCADA networks, we were able to use this traffic to characterize non-malicious SCADA traffic. This characterization was made based on metrics commonly used to characterize worm malware such as interdeparture time of packets and packet sizes. Additional characterization was made from looking at the communication pairs which represent the logical topology inherent in SCADA networks. Three malicious traffic scenarios of increasing detection complexity were then used to evaluate how effective the characterization was at identifying malicious traffic. Common botnets and standard advance persistent threat behaviour proved to be very easy to identify from abnormal communication pairs, unusual interdeprature time and packet sizes which are not present in normal operations. A covert channel based on the DNP3.0 measurement update mechanism proved indistinguishable from normal traffic, but entropy evaluation of this traffic showed that this is explainable by the

similarities with the behaviour of a normal source. This threshold of detection pushes attackers wishing to remain stealthy to a complex method of communication that requires increased tool complexity and extensive reconnaissance to characterize source entropy prior to establishing covert communication. This confirms that anomaly-based detection is effective in restricting the ability of attackers to communicate covertly. The fact that this is done using simple features only strengthens our intuition that SCADA systems provide a favorable environment for the use of this technology.

In summary, we have tested the effectiveness of anomaly-based intrusion detection in SCADA networks. Based on simple features, we have built a model of normal traffic against which common botnets and routine maintenance operations performed by advanced persistent threats easily stand out. To evade this basic surveillance method, attackers may move to employ specifically designed covert channels that match the source entropy of the physical system, which is the electric grid in our case. In traditional corporate networks, it would not have been possible to build such a detector because there is no clear structure in the distribution of communication partners, departure time of packets or packet sizes. This lends credence to our intuition that SCADA protocols impose a structure on network traffic that makes anomaly-based detection more effective in SCADA networks than in traditional networks. This support our strategy of increasing surveillance to limit or detect the covert communication used by advanced persistent threats.

CHAPTER 6 CONCLUSION

With the production of a set of features to perform anomaly-based detection in SCADA networks, we have finally reached our goal of helping power grid operators defend their SCADA networks against advanced persistent threats by better understanding how the behaviour of advanced persistent threats will manifest itself in a SCADA network and developing, based on evidence derived from experiments, new tools and techniques to defeat the expected behaviour.

We started by showing that the true nature of SCADA networks was to serve as a basic control loop for the electric grid. The consequence of this was that any attacker that gained access to the SCADA network could send the grid into any state he wishes. We also showed that, should advanced persistent threats attempt to pursue this goal, current research in SCADA security would not provide significant help. We also saw that experimental approaches currently used are not appropriate to perform experiments in network security in the context of SCADA network.

The first step we tackled to reach our goal, was to develop an attacker model for advanced persistent threat behaviour in SCADA networks that did not necessarily involve causing massive physical damage. We introduced the pinprick attack scenario, our first major contribution, in which it is likely that an attacker will cause small amounts of damage that accumulate over time in order to stay under the radar. From this scenario, we developed a strategy of increasing the capability of surveillance, or boosting the radar so to speak, in order to prevent advanced persistent threats from using this scenario.

To test the capability of our detector, we needed to address the lack of experimental infrastructure suitable for network security. To solve this problem we introduced the ICS sandbox concept, our second major contribution, that uses a hybrid approach combining the high fidelity results of emulation and the scalability and cost reduction of simulation to create an experimental setup able to produce high fidelity network data sets for experimentation. An evaluation of the validity of our experimental approach using industry training sessions and the reproduction of a power engineering experiment were also provided to lend further credence to the results produced by the sandbox.

Finally, we were able to test an implementation of anomaly-based intrusion detection, our third major contribution, using the ICS sandbox. Using only simple features, it was possible to detect command and control traffic in SCADA network and push attackers to use complex covert

channels with limited bandwidth to perform their routine maintenance operations. This attests to the validity of our intuition that anomaly-based detection is particularly effective in SCADA networks, revivifying a defensive technique that was considered ineffective because of its poor performance in typical corporate networks.

The sum of these contributions represents a significant improvement in the defense of SCADA networks against advanced persistent threats, including threats from nation state sponsored intelligence agencies. This contributes to the increased reliability of critical infrastructure, and of the electrical grid in particular, in the face of an increasing number of cyber attacks.

This chapter summarizes the contributions made throughout the course of this research project and specifies the limitations on the scope of these contributions. This chapter also proposes several avenues for future research based on this project's contributions. Section 6.1 presents these elements for the field of advanced persistent threat study. Section 6.2. presents the same elements for experimental research in SCADA network security and section 6.3 covers the same ground for traffic analysis and anomaly-based detection in SCADA networks.

6.1 Behaviour of advanced persistent threats

At the beginning of our research project, very little information was available on advanced persistent threats. In fact, the lack of a major cyber incident involving critical infrastructure was cited as a proof that it was not an issue. Some argued that this was explained by the fact that even the most reckless cyber attacker would think twice about causing major disruptions of critical infrastructure. In a sense, that this level of damage would run contrary to the hacker ethic.

The only scenario which was identified has having a potential for massive disruption was cyber warfare. In this scenario, two countries at war would target each other's critical infrastructure. In that vision, two armies wielding massive denial of service infrastructure would bury the adversary's SCADA system under a flood of packets and whoever had the highest bandwidth won. This scenario had little interest in terms of defense, especially for countries which controlled the bandwidth crossing the border with their adversary. Looking at the sophistication of offensive network security, it seemed unlikely that well funded dedicated attackers, intelligence agencies for example, would be constrained to such a narrow scenario.

Our first contribution was to envision a scenario which did not require a state of active warfare (or close to it) to be realized. Adapting the concept of the spectrums of engagements from the land doctrine, we posited that a cyber attacker could engage in limited forms of engagement other than warfighting if political tensions between two nations increased. These engagements would pursue disruption by accruing a large amount of small damage over a long period of time. These were dubbed pinprick attacks.

This contribution was presented to the strategic community for cyber warfare at the 2010 Conference of cyber conflicts (now CyCon) [36] organized by the NATO Cooperative Cyber Defense Center of Excellence.

After this model was presented to the community, real world events provided a test for the predictions of the pinprick attack model. Stuxnet, the first admitted cyber weapon was discovered in the wild and thoroughly analyzed. Additional operational details were then published in the New York Times. Using all this information, we were able to see that the overall strategy behind the Stuxnet attack was in line with the predictions of our pinprick attack model.

Unfortunately, this represents the extent of the validation we could perform on the model. Being able to get more data points to further prove the validity of our theory would have enhanced our confidence in the model. However, realistically, due to the secretive nature of these programs, it is unlikely that we get to peak again behind the curtains as we did with Stuxnet. Naturally, this would have been unthinkable before it occurred with Stuxnet, so the future may hold more surprises.

This work laid important groundwork for future work which would not have been possible without an attack model. Notably, this offensive strategy, because of its low tempo, requires attackers to establish a presence in the target networks for a very long time. In order to do this without the defenders interfering, stealth is required. This opened the door for a defensive strategy focusing on denying the ability for stealth by increasing surveillance which ultimately proved suited for SCADA networks.

This work also established the ground work for further work in strategic studies. For example, this work was cited in the study of cyber militias. Future developments are also possible. We currently pursue work designed to address the integration of low probability high impact scenarios by adapting current risk analysis techniques to adopt an actor-based approach rather

than a scenario-based approach. Also, using the concept of spectrum of warfare which is a by-product of this contribution, we can analyze the requirements for the pre-positioning of cyber weapons.

6.2 The ICS Sandbox

In order to test the conclusions of the analysis of the advanced persistent threats in SCADA networks, a new experimental approach is required. This section summarizes how the ICS sandbox contributed to our overall research effort by providing the experimental framework required. We start by summarizing the contribution of the ICS sandbox to our research effort and to the community in general. Then, we discuss some of the limitations of the work. Finally, we present future work that was made possible by the introduction of the ICS sandbox.

6.2.1 Contribution

In order to test the defensive strategy that we were led to by the analysis of advanced persistent threat behaviour, we needed an experimental platform. From the study of the literature, we noticed that none of the experimental approaches used provided network data to the level of fidelity we required to test the effectiveness of a surveillance-based strategy. In most cases, the approach did not accurately represent cyber physical systems either because of the lack of interaction between the two components or because insufficient validation detracted from the fidelity of the representation. In other cases, such as limited deployment, the approaches did not provide the scalability necessary to represent a real network. In cases that achieved both scale and good cyber-physical interaction, cost tended to be a problem. Alternatively, the focus was not on producing high fidelity network data, but rather on producing high fidelity electrical data. This lack forced us to develop our own apparatus and methodology for generating data sets.

This methodology represents our second contribution to the community which was presented at the 2013 International Symposium for ICS & SCADA Security [37]. Additionally, because it was possible to perform high-risk experiments, the setup was used to provide training to industry practitioners in order to teach them how to react to real incidents. A final contribution of this experimental method is that, unlike real deployments which are under regulatory constraints to protect the confidentiality of their data, the data sets produced for our experiments can be made publicly available for use by the research community.

Also, this work acts a necessary stepping stone for our strategy of defeating advanced persistent threats in SCADA network. In particular, this allows us to provide evidence of the effectiveness of our defensive strategy of using anomaly-based detectors for surveillance.

6.2.2 Limitations

While our approach has the advantage of being able to reproduce the physical effects without imposing a significant burden in terms of lab space and budget, a number of challenges still remain.

The first challenge is that of repeatability of experimentation. In most situations, we want to be able to repeat an experiment a number of times to prove the statistical validity of the results for the independent variable. We also want to study the impact of model simplifications by analyzing the sensitivity of the results to variation in control variables, such as was demonstrated in Calvet and al. [61]. In practice, most of the SCADA components still need to be configured manually. In particular, the MTU asset database, which is used to determine which equipment should be polled, requires extensive manual configuration. HMI visualization screens also need to be crafted by hand if a human is expected to work with them. While using VMware snapshots for sterilizing the environment makes repeatability for independent variables easy, repeatability for control variables would require modifying the SCADA configuration. Using the xCAT tool, it is possible to craft a number of experimental configurations and run them sequentially for repeatability. However, the production of each of those experimental configurations is very time consuming if it cannot be done programmatically.

Another important challenge is the presence of a synchronization problem, caused by the choice to run scripts on the RLS machines at regular intervals to update the power flow simulation values and measurement point values. If a control point value changes between those intervals, for example as a result of a command sent by an operator to trip a breaker, there will be a delay between the change in the control point's value and the electrical network effects. For drastic changes in values, this can have an impact on the soundness of the DNP3 network traffic because the DNP3 protocol allows for traffic initiated outside of polling sequences by the slaves to report outages. This could also create inconsistencies if a command is sent as a polling request from the MTU within the convergence time of the power flow simulator. A full study of the impact of the choice of discrete time rather than discrete event simulation would be required to evaluate the

impact of the design decision. The synchronization problems can also become more significant when a change affects the value of multiple points across a number of RTUs which may not all update at the same time.

A final challenge with our infrastructure is the availability of standard models to validate this emulation approach, and eventually the proposed security solutions. While there exist some toy models for electrical networks, computer networks and SCADA topologies, there are no models that integrate all three aspects. For example, while standard benchmark models exists for power grid simulation (such as those proposed by the IEEE), these models do not describe the corresponding SCADA infrastructure (i.e. the placement of measurement and control points). The physical SCADA test beds have yet to produce data sets (such as traffic captures on the network component) that could be used to validate our ICS sandbox model. Packet captures from live networks could also be used, but unfortunately critical infrastructure operators are typically reluctant to provide the information, due to confidentiality concerns. However, this problem is common to all ICS security research.

In terms of validation, we are very confident of the fidelity and soundness of the SCADA system. Using emulation with commercial products guarantees that the packets sent on the network will be properly formatted. Going back to the Shannon model, this corresponds to the encoding and decoding boxes. This means that the warden is seeing the correct messages going back and forth on the channel.

Based on our experience with the ICS sandbox for training, we also believe the basic network infrastructure to be representative of real networks. While not necessarily a complete sample, real practitioners found the network architecture to represent accurately the type of problems they are facing themselves. Some variation of the type of network based on the industry was reported. For example, some operators with more geographical distribution have their RTUs on different LANs based on geographical location. However, adapting to these observations only requires that the collection of packet captures be distributed and does not detract from the general validity of the network architecture.

This leaves only the question of the electrical system in terms of validity. The validity of the electrical simulation can be addressed by choosing a simulator which has been vetted by peer-review in its field. This gives us a reasonable expectation of the soundness of results. Even with

peer-reviewed simulators, the simulator may not produce results at a granularity appropriate for our needs. For example, the triggering of a breaker might create transitional effects in the network that may not be captured by a steady state simulator. In that sense, it is important to select a simulator that is appropriate to the kind of experimentation we are running.

Other questions related to the integration also surface. For example, what kind of sensors would a grid operator install in a substation? What is the precision of their measurement? When a breaker is operated, how long does it take for the pneumatic system to fully complete the process? How does that fit with the update lag? A number of other questions like this can shape the entropy of the source and ultimately affect the soundness of traffic. However, these questions are not linked to the validation of the source traffic. Rather, these questions are related to the calibration of the machine.

Measurement tools usually need to be calibrated. For example, a balance giving results accurate to the milligram will produce erroneous results if the zero was not set properly. In the case of the SCADA Sandbox, the tweaking of the granularity of the representation of the electrical source is ultimately a calibration of the measurement device to a specific setting. In the context of building a warden, the fluctuation of the source entropy represents a baseline level of noise in the system. This baseline will vary from one grid operator to another. Some will run systems that are very deliberately configured and hardened while other will build patchwork systems or default configurations. In that sense, the lack of calibration of the Sandbox does not detract from its validity. However, if the Sandbox could be calibrated to an operational production system, it would enhance the confidence we have in the validity of the modelling of the source. Unfortunately, due to the lack of availability of production level SCADA traffic, it is not possible to perform this experiment.

6.2.3 Future work

In terms of future work, the tackling of various limitations of the ICS sandbox present a number of interesting challenges for the software engineering and for the validation communities. In the same vein, the power engineering community could be interested in testing a number of power simulators and their possible integration to the ICS sandbox. However, additional research paths are also opened by the ICS sandbox.

One such path is impact assessment research in the vein of Salmeron et al. [119]. Naturally, the scenarios from Salmeron and al. are not representative of cyber attacks. One of their main underlying assumptions is that the incremental cost of causing more disruptions is significant. So, a terrorist has to make choices in terms of disruption targets to allocate his attack resources adequately. In terms of cyber attacks, the initial cost of intruding in the system is "high" (depending on the security posture of the SCADA network), but the marginal costs of causing more disruption is often zero. For example, if I find a remote exploit that gives me administrative access to an RTU, it is likely that the same exploit will give me the same access to all other RTUs of the same brand and software version. This enables attackers to replicate impact at no cost on a scale undreamt of by physical terrorist. However, the use of this methodology to perform impact assessment for cyber attacks is still relevant in a number of cases. For example, the case where there is a marginal cost to infecting more systems, such as infection by a human carrying a USB key. Impact assessment of indiscriminate cyber attacks where impacts on the electrical grid are either incidental or not pre-planned are also interesting. For example, a denial of service attack that prevents operators from reconfiguring the system after a failure, or a malware that causes breaker to open at random.

With future work, it would be possible to perform more accurate impact assessment. This specific implementation of the ICS sandbox, in addition to suffering from the limitations presented in section 6.2.2, could benefit from some improvements. Most of those improvements would come from using an electrical network simulator with less limitations. The major drawback of PyPower is the validity of the results for interdicted scenarios. Because of the absence of a load shedding model, there are many scenarios where the calculator cannot converge on a solution that fills the constraints. However, PyPower is open source and could be modified to address this issue. A second drawback of PyPower, shared by all steady-state electrical simulators, is that it is not possible to observe transient effects. This restricts the kinds of cyber attack that can be performed. For example, it would be impossible to cause line breaks due to triggering physical protections from a spike in current in a transitory state. Finally, the use of optimized power flow solvers imposes a power generation network discipline that cannot be decoupled from the electrical stimulation. The use of a generator where this is possible could help model more attacks.

Using the ICS sandbox rather than using conventional electrical simulators for impact assessment has many advantages for cyber security research. First, it is possible to test actual malware collected from the wild (small adjustments may be necessary if a command and control server is necessary). This increases the fidelity of the attack scenario. Second, electrical emulators, unlike the ICS sandbox, have no model of the IP structure and instead rely on physical proximity of equipment. Two pieces of equipment that are hundreds of kilometers apart physically may be neighbours on the network. Third, power flow simulators assume that operations can be optimally delivered and assume the availability of perfect data, the testing of the impact of false data from infected SCADA devices cannot be considered.

6.3 Anomaly-based detection in SCADA networks

Using the ICS sandbox as a means to generate high fidelity network data, it was possible to test the conclusions of the analysis of the advanced persistent threat in SCADA networks, i.e. that increased surveillance would restrict the ability of an attacker to remain hidden in a SCADA network for a long period of time. This section summarizes the development of a technique using anomaly-based detection to detect command and control communication in the network. We start by summarizing the contribution of this research. Then, we discuss some of the limitations of the work. Finally, we present future work that can use our findings as the basis for the research.

6.3.1 Contribution

By analyzing the behaviour of advanced persistent threats and proposing the pinprick attack scenario, it was possible to devise a defense based on surveillance. Prior work had been done on detecting attackers in SCADA networks. Unfortunately, a lot of this research was not focused on finding command and control type communications which is the cornerstone of the ability of attackers to persist in the network. Of the research that was able to do so, the majority did not provide validation of their performance. However, a small number of researchers focused on the predictability of SCADA networks to detect attackers. Unfortunately, due to the lack of network traffic data or from a lack of a deep understanding of the behaviour of attackers, this research did not lead to actionable anomaly-based detection.

Our contribution was to take three simple features available without any deep packet inspection and create an anomaly-based intrusion detection system for a SCADA network that detects

command and control channels. The fact that this detector was effective, proves that, while the technology is considered to be unreliable in traditional corporate networks, anomaly-based intrusion detection is effective in SCADA networks because of the regularity of the network traffic.

While none of the characteristics we presented would prevent an attacker from building a tool that would mimic the properties of legitimate traffic, each of those, taken in isolation, could be used to detect malicious activities from common tools. In addition, when taken together, the characteristics we presented create a profile that severely limits the options of even a dedicated attacker using mimicry attacks. It is difficult to do a lot of things when you are limited to sending a small number of packets of specific size at a specified time over a specified network link. In addition, this greatly increases the amount of reconnaissance that attackers are required to perform in order to achieve a high level of stealth. If we take the example of the packet length distribution for covert channels, an attacker would need a good sample of the level of variability of measurement points attached to it if they would want to match the distribution. Additional analysis of the protocol could further tighten the patterns of normality (request packets are followed by a small flurry then a long silence, packet size observations always occur in a specific pattern, etc.), but, using only the easily measurable logical topology, interdeparture time and packet length features, we managed to provide interesting possibilities for detection.

While this method was developed based on our experimental network, it should be applicable to the majority of production level networks. The features we used are the consequences of the protocol definition, and of its use of polling in particular. Most other SCADA protocols, such as Modbus, follow the same design principles and, even though the exact values might differ, will also have the same regularity in terms of distributions. The effort to build these distributions, and evaluate their fitness to act as features for anomaly detection, would mostly reside in adapting the testbed used in this experiment to incorporate Modbus equipment. The main hurdle for this project is the acquisition of Modbus aware SCADA equipment. In that sense, we can argue that the exclusive use of the DNP3.0 protocol does not detract from the validity of the claims.

This contribution allows us to meet our goal of providing new tools and techniques to defeat advanced persistent threats targeting SCADA networks. Our ability to detect common botnet command and control, and the maintenance channel from common hacking tools already

significantly degrades the ability of advanced persistent threats to remain undetected. According to our analysis, the use of covert channels closely mimicking the behaviour of an electrical network is required to avoid detection. This level of sophistication is well out of the reach of most advanced persistent threats actors such as cyber gangs. For adversaries for which the development of covert channels tailored to their target network is within the realm of possibility, such as nation state backed intelligence agencies, their ability to perform routine actions while remaining stealthy is still heavily hindered. First, the use of a covert channel such as the one presented severely restricts their bandwidth. This forces cyber weapon maintenance to longer schedules and limits their ability to respond to defender moves. Second, the need to fully characterize the entropy of the system in order to calibrate their covert channel significantly increases the reconnaissance requirements for mounting a successful attack. For example, if an attacker attempts to move a high volume of data from a measurement point that seldom varies, or that varies with a distribution other than Gaussian noise, the entropy will not match the distribution and the attacker may be detected. Imposing this constraint on the operations of an attacker this advanced represents a significant headway on a problem on which we had little previous traction.

6.3.2 Limitations

The work presented here does suffer from some limitations. The major limitation is the undetermined validity of the ICS sandbox, and of the electrical model in particular. In the absence of publicly available data of live-world SCADA systems, it is not possible to ascertain with certainty that our system behaves as it should. The combination of emulation and simulation as described in section 4.1 does provide a reasonable guarantee that the systems follow the correct protocols, but the system cannot be calibrated. Having detailed knowledge of how real systems are operated would enable us to choose more representative values in terms of numbers of RTU, number of points per RTU and so on. Our sensitivity analysis shows that the choice of these parameters has minimal effects on the results, but validating against a real system would increase the confidence in our results. Alternatively, the successful application of our anomaly detection on a production level system could also provide the same confidence in our results. Unfortunately, there is no publicly available data to test it on.

Finally, it could be argued that the limited choice of features for anomaly detection is a shortcoming of our work. Normally, by increasing the number of features, we would provide even more restrictions for the attacker and thus limit his bandwidth even more. It would even be possible to use data mining to find features that are not obvious to a protocol analysis or to train a machine learning anomaly detector on clean data. However, the production of such a feature list in the face of the limited noise model seems premature. As such, keeping the focus on coarse grained features that provide fairly strong indicators of compromise seems appropriate. Naturally, this could be the topic of future work in the domain.

6.3.3 Future work

In terms of future work enabled by our research, one interesting avenue would be to investigate if further refining our source model would affect our capability of detection. In that sense, the development, in tandem with researchers in power engineering, of a full model of a power grid and its corresponding SCADA network using a real-time simulator able to model transient effects would represent the ultimate source of data to characterize normal SCADA traffic.

Another research axis would be to develop a tool that is able to automatically build the baseline and detect malicious activity. This work using results from our characterization is currently in progress as an undergrad project. The tool could then be provided to an industrial partner to test its effectiveness in a real network deployment without violating confidentiality. The results from this test could further validate our approach.

Other research could be undertaken to develop more features for detection. The use of machine learning approaches could provide us with features that were not previously expected. The use of state machine-based features that further leverage the fact that SCADA traffic distributions are not Markovian (for example, a response packet always has the same size and always follows a query packet of standard size, so there is a memory-based pattern on packet sizes) could also further decrease the wiggle room of attackers. Finally, using deep packet inspection, or partial inspection of packet payloads could allow us to create features based on the DNP3.0 protocol instead of relying purely on TCP and IP headers.

We could also pursue other paradigms for intrusion detection. At first glance, SCADA networks are so regular that white listing packets based on a small number of features could be considered.

For example, it may be possible to create an exhaustive list of SCADA commands and allowed responses to these commands. Typically, these commands and responses will have fixed packet sizes and headers. This could be used to create a list of the possible values for these features and perform intrusion detection with a white list. This approach might require additional research work in rules-based intrusion detection for which tools are not yet built using this paradigm.

Another intrusion detection paradigm that could be tested would be machine learning-based anomaly detection. In particular, it would be interesting to test the feasibility of training the intrusion detector on a production system reproduced in the ICS sandbox and then move the IDS to the real production network. This could provide a method to ensure that machine learning-based intrusion detection does not include prior infections in its baseline for normal traffic.

Overall, the entire research effort, whether we consider the strategic study of advanced persistent threats, the development of tools to perform experimental research or tools for the detection of command and control channels in SCADA networks, represents a first series of contributions to the problem space. The stepping stones laid in the tackling of this research work can be used to address other problems in the larger issue of the security of SCADA networks.

CHAPTER 7 BIBLIOGRAPHY

[1] Riptech, "Understanding SCADA System Secuirty Vulnerabilities," January 2001. [Online]. Available: http://www.iwar.org.uk/cip/resources/utilities/SCADAWhitepaperfinal1.pdf. [Accessed 31 July 2013].

[2] R. Graham and D. Maynor, "SCADA Security and Terrorism: We're not crying wolf," January 2006. [Online]. Available: http://www.blackhat.com/presentations/bh-federal-06/BH-Fed-06-Maynor-Graham-up.pdf. [Accessed 31 July 2013].

[3] K. O'Connel, "INTERNET LAW - CIA Report: Cyber Extortionists Attacked Foreign Power Grid, Disrupting Delivery," Internet Business Law Services, january 2008. [Online]. Available: http://www.ibls.com/internet_law_news_portal_view.aspx?id=1963&s=latestnews. [Accessed 31 July 2013].

[4] S. Jay and M. Miller, "Lessons Learned from the Maroochy Water Breach," in *Critical Infrastructure Protection*, vol. 253, Boston, Springer, 2007, pp. 73-82.

[5] W. Allsop, Unauthorised Access, United Kingdom: Wiley and sons, 2009.

[6] A. Greenberg, "America's Hackable Backbone," 22 August 2008. [Online]. Available: http://www.forbes.com/2007/08/22/scada-hackers-infrastructure-tech-security-cx_ag_0822hack.html. [Accessed 31 July 2013].

[7] Radio-Canada, "Le service est rétabli," 15 December 2009. [Online]. Available: http://www.radio-canada.ca/regions/Quebec/2009/12/15/002-panne_hydro_mardi.shtml. [Accessed 31 July 2013].

[8] Radio-Canada, "Grosse bévue," 19 December 2009. [Online]. Available: http://www.radio-canada.ca/nouvelles/societe/2009/12/16/049-panne-erreur-hydro.shtml. [Accessed 31 July

2013].

[9] C.-W. Ten, C.-C. Liu and M. Govindarasu, "Vulnerability Assessment of Cybersecurity for SCADA Systems Using Attack Trees," in *Power Engineering Society General Meeting*, 2007.

[10] D. E. Nordel, "Terms of Protection," *IEEE Power and Energy magazine*, vol. 10, no. 1, pp. 18-23, 2012.

[11] S. Gold, "The SCADA Challenge: Securing Critical Infrastructure," *Network Security*, vol. 2009, no. 8, pp. 8-10, 2009.

[12] J. Wiles, T. Claypoole, P. Drake, P. A. Henry and L. J. J. Johnson, Techno Security's Guide to Securing SCADA, Burlington, MA: Syngress, 2007.

[13] E. Byres, "The Air Gap: SCADA's Enduring Security Myth," *Communications of the ACM*, vol. 56, no. 8, pp. 29-31, 2013.

[14] CSIS Comission on Cyber Security for the 44th Presidency, "Securing Cyberspace for the 44th presidency," December 2008. [Online]. Available: http://csis.org/files/media/csis/pubs/081208_securingcyberspace_44.pdf. [Accessed 31 July 2013].

[15] R. L. Krutz, Securing SCADA Systems, Indianapolis: Wiley, 2006.

[16] Department of Homeland Security, "The National Strategy to Secure Cyberspace," 2003.

[17] P. Welander, "Cyber Security Hits Home," *Control Engineering*, vol. 56, no. 1, pp. 40-43, 2009.

[18] S. Dynes, E. Goetz and M. Freeman, "Cyber Security: Are Economic Incentives Adequate?," in *Critical Infrastructure Protection*, Boston, Springer, 2007, pp. 15-27.

[19] The Security Incidents Organization, "The Repository of security Incidents," The Security

Incidents Organization, 2013. [Online]. Available: http://www.securityincidents.net/. [Accessed 8 August 2013].

[20] Public Safety Canada, "Canadian Cyber Incident Response Centre (CCIRC)," Government of Canada, 7 August 2013. [Online]. Available: http://www.publicsafety.gc.ca/cnt/ntnl-scrt/cbr-scrt/ccirc-ccric-eng.aspx. [Accessed 8 August 2013].

[21] Department of Homeland Security, "The Industrial Control Systems Cyber Emergency Response Team (ICS-CERT)," United States Gvernment, 2013. [Online]. Available: http://ics-cert.us-cert.gov/. [Accessed 8 August 2013].

[22] N. Falliere, L. O. Murchu and E. Chien, "W32.Stuxnet Dossier Version 1.4," Symantec Security Response, 2011.

[23] D. E. Sanger, "Obama Order Sped Up Wave of Cyberattacks Against Iran," New York Times, 1 June 2012. [Online]. Available: http://www.nytimes.com/2012/06/01/world/middleeast/obama-ordered-wave-of-cyberattacks-against-iran.html?pagewanted=all. [Accessed 8 August 2013].

[24] Byres Security inc., "Using Tofino to control the spread of Stuxnet malware," 2010. [Online]. Available: http://www.mtl-inst.com/images/uploads/datasheets/App_Notes/AN-BYRES119.pdf. [Accessed 27 May 2013].

[25] G. Gritsai, A. Timorin, Y. Goltsev, R. Ilin, S. Gordeychik and A. Karpin, "SCADA Safety in Numbers v1.1*," 2012. [Online]. Available: http://www.ptsecurity.com/download/SCADA_analytics_english.pdf#page=6&zoom=auto,0,310. [Accessed 8 August 2013].

[26] Angencies, " US video shows hacker hit on power grid," China Daily, 27 September 2007. [Online]. Available: http://www.chinadaily.com.cn/world/2007-09/27/content_6139437.htm. [Accessed 8 August 2013].

[27] CBS News, "Cyber War," 13 June 2010. [Online]. Available:

http://www.cbsnews.com/video/watch/?id=6578069n. [Accessed 8 August 2013].

[28] Symantec Security Response, "The Shamoon Attacks," Symantec, 16 August 2012. [Online]. Available: http://www.symantec.com/connect/blogs/shamoon-attacks. [Accessed 8 August 2012].

[29] D. Fisher, "Some Signs Point to Shamoon as Malware in Aramco Attack," Threatpost, 22 August 2012. [Online]. Available: http://threatpost.com/some-signs-point-shamoon-malware-aramco-attack-082212. [Accessed 8 August 2013].

[30] E. Mills, "Virus knocks out computers at Qatari gas firm RasGas," CNET, 30 August 2012. [Online]. Available: http://news.cnet.com/8301-1009_3-57503641-83/virus-knocks-out-computers-at-qatari-gas-firm-rasgas/. [Accessed 8 August 2013].

[31] P. Muncaster, "Iran fingered for attacks on US power firms - Increased levels of online activity have US spooks alert, just a little alarmed," 27 May 2013. [Online]. Available: http://www.theregister.co.uk/2013/05/27/iran_payback_stuxnet_ics_attacks/. [Accessed 8 August 2013].

[32] B. Krebs, "Chinese Hackers Blamed for Intrusion at Energy Industry Giant Telvent," Krebs on Security, 26 September 2012. [Online]. Available: http://krebsonsecurity.com/2012/09/chinese-hackers-blamed-for-intrusion-at-energy-industry-giant-telvent/. [Accessed 8 August 2013].

[33] Mandiant, "APT1 - Exposing One of China's Cyber Espionage Units," February 2013. [Online]. Available: http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf. [Accessed 8 August 2013].

[34] M. Clayton, "Exclusive: Cyberattack leaves natural gas pipelines vulnerable to sabotage," Christian Science Monitor, 27 February 2013. [Online]. Available: http://www.csmonitor.com/Environment/2013/0227/Exclusive-Cyberattack-leaves-natural-gas-pipelines-vulnerable-to-sabotage?nav=87-frontpage-mostViewed. [Accessed 8 August

2013].

[35] N. Perlroth, "Critical Infrastructure Systems Seen as Vulnerable to Attack," New York Times, 17 January 2013. [Online]. Available: http://bits.blogs.nytimes.com/2013/01/17/critical-infrastructure-systems-seen-as-vulnerable-to-attack/?src=rechp&_r=0. [Accessed 8 August 2013].

[36] A. Lemay, J. M. Fernandez and S. Knight, "Pinprick Attacks, A Lesser Included Case," in *Conference on Cyber Conflict Proceedings 2010*, Talinn, 2010.

[37] A. Lemay, J. Fernandez and S. Knight, "An isolated virtual cluster for SCADA network security research," in *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013)*, Leicester, 2013.

[38] U.S. Department of Labor, "Illustrated Glossary : Substations," [Online]. Available: https://www.osha.gov/SLTC/etools/electric_power/illustrated_glossary/substation.html. [Accessed 7 July 2013].

[39] M. Shahidehpour and Y. Wang, "Communication and Control in Electric Power Systems:Applications of Parallel and Distributed Processing," Wiley-IEEE Press , 2003, pp. 26,307-348,.

[40] F. F. Wu, K. Moslehi and A. Bose, "Power Systems Control Centers: Past, Present, and Future," *Proceedings of the IEEE*, vol. 93, no. 11, pp. 1890-1908, 2005.

[41] Nematron, "Nematron OptiLogic Series - Flexible Modular System for Cost-Effective Ethernet I/O," 2009. [Online]. Available: http://www.nematron.com/products/legacy/optilogic.html. [Accessed 22 July 2013].

[42] M. D. Schwartz, J. Mulder, J. Trent and W. D. Atkins, "Control System Devices: Architectures and Supply Channels Overview," Sandia National Laboratories, Albuquerque, 2010.

[43] Wikimedia commons, "Siemens Simatic S7-416-3.jpg," 21 June 2010. [Online]. Available: https://en.wikipedia.org/wiki/File:Siemens_Simatic_S7-416-3.jpg. [Accessed 22 July 2013].

[44] G. Clark and D. Reynders, Pratcical Modern SCADA Protocols: DNP3, IEC 60870.5 and Related Systems, China: Newnes (Elsevier), 2008.

[45] U.S. Department of Energy, "National SCADA Test Bed Enhancing control systems security in the energy sector," 16 September 2009. [Online]. Available: http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/NSTB_Fact_Sheet_FINAL_09-16-09.pdf. [Accessed 02 April 2013].

[46] Hydro-Québec, "Rapport annuel 2012," 2012. [Online]. Available: http://www.hydroquebec.com/publications/fr/rapport_annuel/pdf/rapport-annuel-2012.pdf. [Accessed 2 April 2013].

[47] R. J. O'Harrow, "CyberCity allows government hacker to train for attacks," 26 November 2012. [Online]. Available: http://www.washingtonpost.com/investigations/cybercity-allows-government-hackers-to-train-for-attacks/2012/11/26/588f4dae-1244-11e2-be82-c3411b7680a9_story.html. [Accessed 20 March 2013].

[48] G. Dondossola, G. Deconinck, F. Garrone and H. Beitollahi, "Testbeds for Assessing Critical Scenarios in Power Systems Control," in *Critical Information Infrastructure Security*, Rome, 2008.

[49] G. Dondossola, "Testbeds for Assessing Critical Scenarios in Power Control Systems," *Critical Information Infrastructure Security,* vol. 5508, pp. 223-234, 2009.

[50] C. Queiroz, A. Mahmood, J. Hu, Z. Tari and X. Yu, "Building a SCADA Security Testbed," in *hird International Conference on Network and System Security*, 2009.

[51] T. H. Morris, R. B. Vaughn and E. Sitnikova, "Advances in the Protection of Critical Infrastructure by Improvement in Industrial Control System Security," in *Proceedings of*

the Eleventh Australasian Information Security Conference (AISC2013), Adelaide, 2013.

[52] A. Hahn, M. Govindarasu, S. Sridhar, B. Kregel, J. Fitzpatrick, M. Higdon and R. Adnan, "Development of the PowerCyber SCADA Security Testbed," in *CSIIRW '10, Cyber Security and Information Intelligence Research Workshop*, Oak Ridge, 2010.

[53] T. H. Kobayashi, A. B. Batista, J. P. S. Medeiros, J. M. F. Filho, A. J. M. Brito and P. S. Motta Pires, "Analysis of Malicious Traffic in Modbus/TCP Communications," *Lecture Notes in Computer Science*, vol. 5508, pp. 200-210, 2009.

[54] A. Giani, B. Sinopoli, A. Shah, G. Karsai and J. Wiley, "The TRUST-SCADA Experimental Testbed: Design and Experiments," in *TRUST 2008 Autumn Conference*, Nasville, 2008.

[55] W. Chunlei, F. Lan and D. Yiqi, "A Simulation Environment for SCADA Security Analysis and Assessment," in *2010 International Conference on Measuring Technology and Mechatronics Automation*, Changsha, 2010.

[56] C. Queiroz, A. Mahmood and Z. Tari, "SCADASim—A Framework for Building SCADA Simulations," *IEEE TRANSACTIONS ON SMART GRID*, vol. 2, no. 4, pp. 589-597, 2011.

[57] C. M. Davis, J. Tate, H. Okhravi, C. Grier, T. Overbye and D. Nicol, "SCADA Cyber Security Testbed Development," in *38th North American Power Symposium, 2006. NAPS 2006*, Carbondale, IL., 2006.

[58] D. C. Bergman, D. Jin, D. M. Nicol and T. Yardley, "The Virtual Power System Testbed and Inter-Testbed Integration," in *2nd Workshop on Security Experimentation and Test (CSET2009)*, Montreal, 2009.

[59] R. Braden, D. Kim, C. C. Neuman, A. D. Joseph, K. Sklower, R. Ostrenga and S. Schwab, "Experience with DETER: a testbed for security research," in *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006*, Barcelona, 2006.

[60] M. Hibler, R. Ricci, L. Stoller, J. Duerig, S. Guruprasad, T. Stack, K. Webb and J. Lepreau, "Large-scale Virtualization in the Emulab Network Testbed," in *USENIX 2008 Annual Technical Conference*, Boston, MA, 2008.

[61] J. Calvet, C. R. Davis, J. M. Fernandez, W. Guizani, M. Kaczmarek, J.-Y. Marion and P.-L. St-Onge, "Isolated virtualised clusters:testbeds for high-risk security experimentation and training," in *Proceedings of the 3rd international conference on Cyber Security experimentation and test. CSET 2010*, Berkeley, CA, 2010.

[62] I. N. Fovino, A. Carcano, M. Masera and A. Trombetta, "An experimental investigation of malware attacks on SCADA systems," *Infrastructure Protection*, vol. 2, no. 4, pp. 139-145, 2009.

[63] S. Sridhar and G. Manimaran, "Data Integrity Attacks and their Impacts on SCADA Control System," in *2010 IEEE Power and Energy Society General Meeting*, Minneapolis, 2010.

[64] A. Bobbio, G. Bonanni, E. Ciancamerla, R. Clemente, A. Iacomini, M. Minichino, A. Scarlatti, R. Terruggia and E. Zendri, "Unavailability of critical SCADA communication links interconnecting a power grid and a Telco network," *Reliability Engineering & System Safety*, vol. 95, no. 12, p. 1345–1357, 2010.

[65] V. Pothamsetty and M. Franz, "SCADA HoneyNet Project," 15 July 2005. [Online]. Available: http://scadahoneynet.sourceforge.net/. [Accessed 31 July 2013].

[66] HoneyNet Project, "The HoneyNet Project," July 2013. [Online]. Available: http://www.honeynet.org/. [Accessed 31 July 2013].

[67] T. Simonite, "Chinese Hacking Team Caught Taking Over Decoy Water Plant," 2 August 2013. [Online]. Available: http://www.technologyreview.com/news/517786/chinese-hacking-team-caught-taking-over-decoy-water-plant/. [Accessed 10 October 2013].

[68] D. B., "Formal Modeling and Analysis of the ModBus Protocol," in *Critical Infrastructure Protection*, Boston, Springer, 2007, pp. 189-204.

[69] J. Edmond, M. Papa and S. Shenoi, "Security Analysis of Multilayer SCADA Protocols: a Modbus TCP Case Study," in *Critical Infrastructure* , Boston, Springer, 2007, pp. 205-221.

[70] J. T. Hagen and B. E. Mullins, "TCP veto: A novel network attack and its Application to SCADA protocols," in *Innovative Smart Grid Technologies (ISGT), 2013 IEEE PES* , Washington, 2013.

[71] S. East, J. Butts, M. Papa and S. Shenoi, "A Taxonomy of Attacks on the DNP3 Protocol," in *Critical Infrastructure Protection III*, Boston, Springer, 2009, pp. 67-81.

[72] B. Zhu, A. Joseph and S. Sastry, "A Taxonomy of Cyber Attacks on SCADA Systems," in *2011 IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing*, Dalian, 2011.

[73] Y. Wang, "sSCADA: securing SCADA infrastructure communications," *International Journal of Communication Networks and Distributed Systems,* vol. 6, no. 1, pp. 59-78, 2011.

[74] W. Gao, T. Morris, B. Reaves and D. Richley, "On SCADA control system command and response injection and intrusion detection," in *2010 eCrime Researchers Summit (eCrime)*, Dallas, 2010.

[75] C. Bellitini and J. Rrushi, "Vulnerability Analysis of SCADA Protocol Binaries through Detection of Memory Access Taintedness," in *Proceedings of the 2007 Workshop on Information Assurance*, West Point, NY, 2007.

[76] G. Deravajan, "Unraveling SCADA Protocols: Using SULLEY Fuzzer," in *Defcon 15*, Las Vegas, 2007.

[77] A. Carcano, I. N. Fovino, M. Masera and A. Trombetta, "Scada Malware, a Proof of

Concep," *Critical Information Infrastructure Security Lecture Notes in Computer Science* , vol. 5508, pp. 211-222, 2009.

[78] M. Davis, "Smartgrid Device Security: Adventures in a New Medium," in *Black Hat, U.S.A.*, Las Vegas, 2009.

[79] P. McDaniel and S. McLaughlin, "Security and Privacy Challenges in the Smart Grid," *IEEE Security & Privacy,* vol. 7, no. 3, pp. 75-77, 2009.

[80] A. K. Wright, J. A. Kinast and J. McCarthy, "Low-Latency Cryptographic Protection for SCADA Communications," *Applied Cryptography and Network Security, Lecture notes in computer science,* vol. 3089, pp. 263-277, 2004.

[81] M. Majdalawieh, F. Parisi-Presicce and D. Wijesekera, "DNPSec: Distributed Network Protocol Version 3 (DNP3) Security Framework," in *Advances in Computer, Information, and Systems Sciences, and Engineering, Proceedings of IETA*, 2005.

[82] DNP3 Users Group Technical Committee, "DNP3 Secure Authentication Specification Version 2.0," DNP3 Users Group, 2008.

[83] S. C. Patel, G. D. Bhatt and J. H. Graham, "Improving the Cyber Security of SCADA Communication Networks," *Communications of the ACM,* vol. 52, no. 7, pp. 138-142, 2009.

[84] S. B. Prabhakar, S. Bagaria and Z. Saquib, "Flexi-DNP3 : Flexible Distributed Network Protocol Version 3 (DNP3) for SCADA security," in *2011 International Conference on Recent Trends in Information Systems*, Kolkata, 2011.

[85] G. M. Coates, K. M. Hopkinson, S. R. Graham and S. H. Kurkowski, "A Trust System Architecture for SCADA Network Security," *IEEE Transactions on Power Delivery,* vol. 25, no. 1, pp. 158-169, 2010.

[86] R. Dawson, C. Boyd, E. Dawson and J. M. Gonzalez Nieto, "SKMA: a key management

architecture for SCADA systems," in *ACM International Conference Proceeding Series*, 2007.

[87] L. Piètre-Cambacédès and P. Sitbon, "Cryptographic Key Management for SCADA Systems-Issues and Perspectives," in *Proceedings of the 2008 International Conference on Information Security and Assurance (ISA 2008)*, Busan, 2008.

[88] O. Pal, S. Saiwan, P. Jain, Z. Saquib and D. Patel, "Cryptographic Key Management for SCADA System: An Architectural Framework," in *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, Trivandrum, 2009.

[89] A. Rezai, P. Keshavarzi and Z. Moravej, "Secure SCADA communication by using a modified key management scheme," *ISA Transactions*, vol. 52, no. 4, pp. 517-524, 2013.

[90] J. Pollet, "patriotSCADA Distributed Firewall for SCADA and Industrial Networks," plantData Technology, [Online]. Available: http://www.controlglobal.com/whitepapers/wp_001_SCADApollet.pdf. [Accessed 31 July 2013].

[91] Tofino Security, "Tofino," 2013. [Online]. Available: http://www.tofinosecurity.com/. [Accessed 4 June 2013].

[92] R. Bradetich and P. Oman, "Connecting SCADA Systems to Corporate IT Networks Using Security-Enhanced Linux," 12 September 2007. [Online]. Available: http://www2.selinc.com/techpprs/6289_SecurityLinux_RB_20070912.pdf. [Accessed April 2010].

[93] H. Hadeli, R. Schierholz, M. Braendle and C. Tuduce, "Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration," in *IEEE Conference on Emerging Technologies & Factory Automation, 2009. ETFA 2009*, Mallorca, 2009.

[94] Z. Anwa, M. Montanaria, A. Gutierreza and R. H. Campbel, "Optimal Budget constrained security hardening off control networks for critical cyber-infrastructures," *International Newspaper of Critical Infrastructure Protection,* vol. 2, no. 1-2, pp. 13-25, 2009.

[95] D. Peterson, "Intrusion Detection and Cyber Security Monitoring of SCADA and DCS Networks," in *ISA Automation West*, Long Beach, CA, 2004.

[96] D. Yang, E. Usynin and J. W. Hynes, "Anomaly-Based Intrusion Detection for SCADA Systems," 2006. [Online]. Available: http://entrac.iaea.org/I-and-C/TM_IDAHO_2006/CD/IAEA%20Day%202/Hines%20paper.pdf. [Accessed April 2010].

[97] P. Oman and M. Phillips, "Intrusion Detection and Event Monitoring in SCADA Networks," in *Critical Infrastructure Protection,* Boston, Springer, 2007, pp. 161-173.

[98] J. Bigham, D. Gamez and N. Lu, "Safeguarding SCADA Systems with Anomaly Detection," *Computer Network Security, lecture Notes in Computer Science,* vol. 2776, pp. 171-182, 2003.

[99] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino and A. Trombetta, "A Multidimensional Critical State Analysis for Detecting Intrusions in SCADA Systems," *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS,* vol. 7, no. 2, pp. 179-186, 2011.

[100] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner and A. Valdes, "Using model-based intrusion detection for SCADA networks," in *Proceedings of the SCADA Security Scientific Symposium*, Miami Beach, 2007.

[101] N. Goldenberg and A. Wool, "Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems," *International Journal of Critical Infrastructure Protection,* vol. 6, no. 2, pp. 63-75, 2013.

[102] F. Schuster, A. Paul and H. Konig, "Towards Learning Normality for Anomaly Detection in Industrial Control Networks," in *7th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security, AIMS 2013*, Barcelona, 2013.

[103] D. A. Rusu, B. Genge and C. Siaterlis, "A Systematic Approach for Connection Pattern-based Anomaly Detection in SCADA systems," in *The 7th International Conference Interdisciplinarity in Engineering (INTER-ENG 2013)*, Tg. Mures, 2013.

[104] M. V. Mahoney and P. K. Chan, "An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection," in *Recent Advances in Intrusion Detection*, Heidelberg, 2003.

[105] J. Langill, "Implementing a Network-based Intrusion Detection System for Control Systems," in *ICSJWG 2011 Fall Conference*, 2011.

[106] R. R. R. Barbosa, R. Sadre and A. Pras, "Towards Periodicity Based Anomaly Detection in SCADA Networks," in *EEE 17th Conference on Emerging Technologies & Factory Automation, ETFA 2012*, Krakow, 2012.

[107] R. R. R. Barbosa, A. Pras and R. Sadre, "Flow whitelisting in SCADA networks," in *Seventh Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection*, Washington, 2013.

[108] SelectUSA, "The Energy Industry in the United States," U.S. Department of Commerce, [Online]. Available: http://selectusa.commerce.gov/industry-snapshots/energy-industry-united-states. [Accessed 17 September 2013].

[109] R. A. Falkenrath, "Analytic Models and Policy Prescription:," in *Belfer Center for Science and International Affairs - Executive Session on Domestic Preparedness*, Harvard University, 2000.

[110] National Defence Canada, "Canada's Army," 1998.

[111] National Defence Canada, "Conduct of Land Operations," 1998.

[112] A. J. Tellis, J. Bially, C. Layne and M. McPherson, Measuring National Power in the Postindustrial Age, Santa Monica: RAND Corporation, 2000.

[113] R. W. Smith and S. Knight, "Predictable Design of Network-Based Covert Communication Systems," in *2008 IEEE Symposium on Security and Privacy*, Oakland, 2008.

[114] General Electric, "GENe Software Suite," January 2011. [Online]. Available: http://www.gedigitalenergy.com/products/brochures/uos/GENeSoftwareSuite.pdf. [Accessed 8 April 2013].

[115] B. Rosset, "Rapport de Stage - Étape 5," Polytech Paris Sud, Paris, 2012.

[116] IEEE Reliability Test System Task Force, "IEEE Reliability Test System," *IEEE Transactions on Power Apparatus and Systems,* Vols. PAS-98, no. 6, pp. 2047-2054, Nov./Dec. 1979.

[117] IEEE Reliability Test System Task Force of Applications of Probability Methods Subcommittee, "IEEE reliability test system-96," *IEEE Transactions on Power Systems,* vol. 14, no. 3, pp. 1010-1020, 1999.

[118] R. Bacher, "Optimization in Planning and Operation of Electric Power Systems," *Physica-Verlag (Springer),* pp. 217-264, May 1993.

[119] J. Salmeron, K. Wood and R. Baldick, "Analysis of Electric GridSecurity Under Terrorist Threat," *IEEE Transactions on power systems,* vol. 19, no. 2, pp. 905-912, 2004.

[120] R. Lincoln, "GitHub repository for rwl/PYPOWER," GitHub, November 2012. [Online]. Available: https://github.com/rwl/PYPOWER. [Accessed 01 October 2013].

[121] A. A. Cardenas, S. Amin and S. Sastry, "Research Challenges for the Security of Control Systems," in *3rd USENIX Workshop on Hot Topics in Security (Hotsec 2008)*, San Jose,

2008.

[122] R. R. Barbosa, R. Sadre and A. Pras, "Difficulties in Modeling SCADA Traffic: A Comparative Analysis," in *Passive and Active Measurement: 13th International Conference, PAM 2012*, Vienna, 2012.

[123] M. Crotti, M. Dusi, F. Gringoli and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *ACM SIGCOMM Computer Communication Review, Volume 37 Issue 1*, pp. 5-16, January 2007.

[124] G. Gu, R. Perdisci, J. Zhang and W. Lee, "BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection," in *17th Usenix Security Symposium - Usenix Security '08*, San Jose, 2008.

[125] F. Erman, M. Arlitt and A. Mahanti, "Traffic classification using clustering algorithms," in *MineNet '06 Proceedings of the 2006 SIGCOMM workshop on Mining network data*, Pisa, 2006.

[126] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *SIGMETRICS '05 Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, Banff, 2005.

[127] N. William, S. Zander and A. Grenville, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *ACM SIGCOMM Computer Communication Review, Volume 36 Issue 5*, pp. 5-16, October 2006.

[128] C. Wright, F. Monrose and G. M. Masson, "HMM profiles for network traffic classification," in *VizSEC/DMSEC '04 Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, 2004.

[129] A. Dainotti, A. Pescapé and G. Ventre, "Worm Traffic Analysis and Characterization," in *The IEEE International Conference on Communications 2007 - ICC-2007* , Glasgow, 2007.

[130] College of Saint Benedict and Saint John's University, "KS-test Data Entry," [Online]. Available: http://www.physics.csbsju.edu/stats/KS-test.n.plot_form.html. [Accessed 9 September 2013].

[131] B. Stock, J. Gobel, M. Engelberth, F. C. Freiling and T. Holz, "Walowdac - Analysis of a Peer-to-Peer Botnet," in *2009 European Conference on Computer Network Defense*, Milan, 2009.

[132] J. Calvet, C. R. Davis and P.-M. Bureau, "Malware Authors don't learn and that's good," in *Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on* , Montreal, 2009.

[133] J. Calvet, C. R. Davis, J. M. Fernandez, J.-Y. Marion, P.-L. St-Onge, W. Guizani, B. Pierre-Marc and A. Somayaji, "The case for in-the-lab botnet experimentation: creating and taking down a 3000-node botnet," in *ACSAC '10 Proceedings of the 26th Annual Computer Security Applications Conference* , New York, NY, 2010.

[134] Rapid7, "Metasploit Framework," 2013. [Online]. Available: http://www.metasploit.com/. [Accessed 9 September 2013].

[135] Vice-présidence Distribution Direction Plans et Stratégies d'affaires Orientations du réseau, "Characteristics and target values of the voltage supplied by the Hydro-Québec medium and low voltage systems," Hydro Québec, 5 June 2001. [Online]. Available: http://www.hydroquebec.com/pdf/en/qualite_tension.pdf. [Accessed 9 September 2013].

[136] S. Gianveccio and H. Wang, "Detecting covert timing channels: an entropy-based approach," in *CCS '07 Proceedings of the 14th ACM conference on Computer and communications security*, Alexandria, 2007.

[137] K. E. Reinhard, P. W. Sauer and A. D. Dominguez-Garcia, "On Computing Power System Steady-State Stability Using Synchrophasor Data," in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, IEEE, 2013.

[138] Wikimedia commons, "File:Feedback loop with descriptions.svg," 15 October 2008. [Online]. Available: http://upload.wikimedia.org/wikipedia/commons/thumb/2/24/Feedback_loop_with_descriptions.svg/1000px-Feedback_loop_with_descriptions.svg.png. [Accessed 17 July 2013].

[139] J. Hull, H. Khurana, T. Markham and K. Staggs, "Staying in Control: Cybersecurity and the Modern Electric Grid," *IEEE Power and Energy Magazine,* pp. 41-48, Jan.-Feb. 2012.

[140] Y. Y. Haimes, B. M. Horowitz, J. H. Lambert, J. R. Santos, C. Lian and K. G. Crowther, "Inoperability Input-Output Model forInterdependent Infrastructure Sectors. I: Theory and Methodology," *ASCE Journal of Infrastructure Systems,* vol. 11, no. 2, pp. 67-79, 2005.

[141] Y. Haimes, J. Santos, K. Crowther, M. Henry, C. Lian and Z. Yan, "Risk Analysis in Interdependent Infrastructures," in *Critical Infrastructure Protection*, Boston, Springer, 2007, pp. 297-310.

[142] Z. Z. Aung and K. Watanabe, "A Framework for Modeling Interdependencies in Japan's Critical Infrastrucure," in *Critical Infrastructure Protection III*, Boston, Springer, 2009, pp. 243-257.

[143] B. Robert and L. Morabito, "The operational tools for managing physical interdependencies among critical infrastructures," *International Journal of Critical Infrastructures,* vol. 4, no. 4, pp. 353-367, 2008.

[144] NERC (North American Energy Reliability Corporation), "CIP Standards," 2013. [Online]. Available: http://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx. [Accessed 31 July 2013].

[145] K. Stouffer, J. Falco and K. Kent, "Recommendations of the National Institute of Standards and Technology, NIST Special publication 800-82," NIST, 2006.

[146] M. A. McQueen, W. F. Boyer, M. A. Flynn and G. A. Beitel, "Quantitative Cyber Risk Reduction Estimation Methodology for a Small SCADA Control," in *Proceedings of the*

39th Hawaii International Conference on System Sciences, Kauai, 2006.

[147] G. N. Ericsson, "Information Security for Electric Power Utilities (EPUs)—CIGRÉ Developments on Frameworks, Risk Assessment, and Technology," *IEEE Transactions on Power Delivery*, vol. 24, no. 3, pp. 1174-1181, 2009.

[148] M. Hentea, "Improving Security for SCADA Control Systems," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 3, pp. 73-86, 2008.

[149] S. C. Patel and Y. Yu, "Analysis of SCADA Security Models," *International Management Review*, vol. 3, no. 2, pp. 68-76, 2007.

[150] A. Cardenas, T. Roosta and S. Sastry, "Rethinking security properties, threat models, and the design space in sensor networks: a case study in SCADA systems," *Ad hoc Networks*, vol. 7, pp. 1434-1447, 2009.

[151] R. Klein, "Information modelling and simulation in broad interdependent critical infrastructure in IRIIS," in *Critical Information Infrastructure Security*, Rome, 2008.

[152] S. M. Papa, W. D. Casper and S. Nair, "Availability Based Risk Analysis for SCADA Embedded Computer Systems," in *The 2011 World Congress in computer Science Computer Engineering and Applied Computing*, Las Vegas, 2011.

[153] A. Nicholson, S. Webber, S. Dyer, T. Patel and H. Janicke, "SCADA security in the light of Cyber-Warfare," *Computers & Security*, vol. 31, no. 4, p. 418–436, 2012.

[154] Public Safety Canada, "National Strategy for Critical Infrastructure," Government of Canada, 7 August 2013. [Online]. Available: http://www.publicsafety.gc.ca/cnt/rsrcs/pblctns/srtg-crtcl-nfrstrctr/index-eng.aspx. [Accessed 8 August 2013].

[155] J. M. Fernandez, "RSA 2013 presentation," 26 February 2013. [Online]. Available: http://secsi.polymtl.ca/rsa2013/RSA-SCADA-JMF.pdf. [Accessed 3 June 2013].

[156] G. Clarke and D. Reynders, "Chapter 6: Advanced Considerations of DNP3," in *Practical Modern SCADA Protocols DNP3, IEC 60870.5 and Related Systems*, Newnes, 2008, pp. 143-169.

[157] P. Bradford, J. Kline, D. Plonka and A. Ron, "A signal analysis of network traffic anomalies," in *IMW '02 Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment* , New York, 2002.

[158] "Malc0de Database," [Online]. Available: http://www.malc0de.com/database/. [Accessed 26 May 2013].

[159] D. Paré, G. Turmel, J.-C. Soumagne, V. Q. Do, S. Casoria, M. Bisonnette, B. Marcoux and D. McNabb, "Validation Tests of The Hypersim Digital Real Time Simulator with a Large AC-DC Network," in *International Conference on Power System Transients - IPST 2003*, New Orleans, 2003.