EXECUTING CYBER SECURITY ATTACKS ON A SMART GRID TESTBED

by

Olaoluwa Olayokun
Bachelor, Bells University of Technology, 2013

A Project Report Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF ENGINEERING

in the Department of Electrical & Computer Engineering

© Olaoluwa Olayokun, 2016
University of Victoria

All rights reserved. This project report may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.
EXECUTING CYBER SECURITY ATTACKS ON A SMART GRID TESTBED

by

Olaoluwa Olayokun
Bachelor, Bells University of Technology, 2013

Supervisory Committee

Dr. Issa Traore (Department of Electrical & Computer Engineering)
Supervisor

Dr. Ashoka Bhat (Department of Electrical & Computer Engineering)
Departmental Member
ABSTRACT

Supervisory Committee

Dr Issa Traore (Department of Electrical & Computer Engineering)
Supervisor

Dr. Ashoka Bhat (Department of Electrical & Computer Engineering)
Departmental Member

Smart Grids have emerged as a very crucial platform for providing timely, efficient, and uninterrupted power supply to consumers. Communication networks in smart grid bring increased connectivity with increased severe security vulnerabilities and challenges. Smart grid can be a prime target for cyber attack because of its critical nature. As a result, smart grid security is already getting a lot of attention from governments, energy industries, and consumers. The threat of malicious attacks against the security of the Smart Grid infrastructure cannot be overlooked. In this project we created a testbed to simulate attacks on a smart grid power distribution environment. This allows studying the impact and extent of damage an attack can cause to a grid, and provides a platform to investigate, in future research, adequate empirical protection models and tools for smart grid.
# TABLE OF CONTENTS

Supervisory Committee ........................................................................................................... 2  
Abstract ......................................................................................................................................... 3  
Table of Contents .......................................................................................................................... 4  
List of Figures .................................................................................................................................. 5  
Acknowledgments ............................................................................................................................ 6  
Dedication ........................................................................................................................................ 7  
1. Introduction ................................................................................................................................... 8  
2. Related Works ............................................................................................................................... 10  
   2.1 Real Hardware Testbed Approach ......................................................................................... 10  
   2.2 Software Simulation Approach ............................................................................................ 10  
3. Smart Grid Network Security and Simulator .................................................................................. 12  
   3.1 Features of Smart Grid Networks ......................................................................................... 12  
   3.2 Smart Grid Network Simulator: The Score Application ...................................................... 12  
   3.3 Smart Grid Security Requirements and Objectives ............................................................. 16  
   3.4 Smart Grid Attacks ................................................................................................................ 19  
4. Smart Grid Security Testbed ......................................................................................................... 22  
   4.1 Testbed Requirements ............................................................................................................ 22  
   4.2 Testbed Configuration and Setup ......................................................................................... 23  
   4.3 Attacks on the Smart Grid Testbed ....................................................................................... 27  
5. Conclusion ....................................................................................................................................... 32  
   Appendix ....................................................................................................................................... 33  
   Reference ....................................................................................................................................... 35
# LIST OF FIGURES

Figure 1: SCORE Architecture .................................................................................................................. 14
Figure 2: Evaluating the risks in smart grid systems.................................................................................. 17
Figure 3: The CIA triad for smart grid security systems ............................................................................. 17
Figure 4: Running SCORE .......................................................................................................................... 23
Figure 5: Smart Grid Power Distribution Network Set-Up on Target Machine ............................................ 24
Figure 6: Screenshot showing the Throughput on each link when running .................................................. 25
Figure 7: Showing the connectivity test between all nodes in the system ................................................... 26
Figure 8: Fisheye Topology View of the Smart Grid Network ................................................................ 27
Figure 9: Nmap Output .............................................................................................................................. 28
Figure 10: DoS command using hping3 ..................................................................................................... 28
Figure 11: Screenshot of the DoS attack ..................................................................................................... 29
Figure 12: A ping from a node to the target during the DoS attack .............................................................. 30
ACKNOWLEDGMENTS

I am really thankful to God for giving me the grace to complete this project within the time frame set. This Project would not have been completed without the guidance of my supervisor, Dr. Issa Traore. I would also love to acknowledge my parents; Olanrewaju and Oluwatoyin Olayokun for their physical, financial and spiritual support during my post graduate study. Lastly, I would like to appreciate my friends and colleagues for their continuous support during the course of this project.
DEDICATION

I would like to dedicate this work to my siblings; Dara, Moyo and Olamide for the constant support and love they have for me.
CHAPTER ONE

INTRODUCTION

By using the literal definition, a smart grid can be defined as the integration of Information Communication Technology into Power Network using both electrical and information communication layer. Even though the use of smart grid has been increasing, there is still no agreed unique definition. However, we take into consideration two main different definitions, one provided by the European Technology Platform [1] and the other one from the National Institute of Standards and Technology (NIST) [2]. The European definition of smart grid is:

“A smart grid is an electricity network that can intelligently integrate the actions of all users connected to it - generators, consumers and those that do both - in order to efficiently deliver sustainable, economic and secure electricity issues.” [3]

This definition is more oriented towards the actors involved in the power network, while the American definition is more oriented towards the technical specificities of the smart grid:

“. . . advanced power grid for the 21st century include the addition and integration of many varieties of digital computing and communication technologies and services with the power delivery infrastructure. Bidirectional flows of energy and two-way communication and control capabilities will enable an array of new functionalities and applications that go well beyond ‘smart’ meters for homes and businesses.” [4]

Despite the little differences in the smart grid definitions, both parties agree on most of the benefit in using smart grid, which include:

- Increase quality and power reliability, which affect not only normal operation, but also allow refining the grid resilience to disruption caused by natural disasters and attacks.
- Provide the users with energy usage information, allowing the implementation of an energy awareness system.
- Allow a more active role of the consumer, increasing their choices by enabling new products, services, and markets.
- Increase environmental benefits and reduce greenhouse emissions by enabling EV and RES integration.
- Preventive maintenance through the continuous power network monitoring system.

A smart grid is a critical infrastructure network with very stringent dependability requirements. The resilience of such network to disasters, both natural and man-made is crucial. Cyber security attacks are among the man-made disasters facing smart grid network.
Understanding how such attacks operate is an important step in designing adequate protection strategies against the underlying threats. The purpose of this project is to set up an attack platform, which allows researchers to execute and study the effect of various attack scenarios against a smart grid simulation platform.

The rest of the report is structured as follows. Chapter 2 summarizes and discusses related work. Chapter 3 gives an overview of smart grid network security requirements and presents the simulation platform used in this project. Chapter 4 presents the smart grid security testbed and attacks to the testbed. Chapter 5 makes some concluding remarks.
CHAPTER TWO

RELATED WORKS

Creating test platform for cyber-physical analysis in Smart Grid is challenging and it has been studied for years. The approaches technically used to solve this issue generally breaks down into two categories: real hardware testbed and software simulation. The approach used in this project is software simulation.

2.1 Real Hardware Testbed Approach

Real hardware testbeds are further divided into two categories: flat-out hardware platforms and hardware in the loop platforms.

2.1.1 Flat-Out Hardware Platform

Flat-out hardware platforms are the ones consisting of total hardware devices. The Korean government selected the whole Jeju Island to build the Smart Grid testbed to allow the testing of Smart Grid technologies and business models [5]. A Renewable Energy Laboratory in Greece was created to set up a central-controlled microgrid testbed which had PV-panels, battery banks and inverters to investigate the proposed Smart Grid topologies [6]. Sensorweb Reserach Laboratory from Georgia State University designed SmartGridLab testbed to test the distributed demand response algorithm. It includes intelligent power switch, power generator, renewable energy sources, smart appliances, and power meter [7].

2.1.2 Hardware-In-The-Loop Platform

Hardware-in-the-loop platform are the ones that have a mixture of both hardware devices and software simulators to achieve the cyber physical analysis of Smart Grid. Hahn et al. in [8] employ devices like Programmable Logic Units (PLUs) and Intelligent Electronic Devices (IEDs) for communication networks and Real-Time Digital Simulators for power network simulation.

2.2 Software Simulation Approach

The software simulation applications for Smart Grid cyber physical analysis can be further divided into two categories: individual simulation platforms and co-simulation platforms.

2.2.1 Individual Simulation Platforms

Individual simulation platforms are those which bring together the simulation features for Smart Grid into one entity. These types of simulations usually aim at and focus on one particular area of interests for Smart Grid. In 2008, Guo et al. designed and created an energy demand management simulator (EDMS) to calculate the response from different deployment strategies of distributed domestic energy management [9].
In 2009, Molderink et al. created from scratch a simulation environment to analyze and control algorithms for energy efficiency [10]. In the simulation created, micro generators, energy buffers and appliances were all modeled and different energy streams like heat and gas were studied.

In 2012, Narayan et al. presented GridSpice [11] a cloud based simulation package for Smart Grid. Leveraging the powerful component of Gridlab-D and Matpower, GridSpice was developed with the main purpose of modeling the interactions between all parts of the electrical network, including generation, transmission, distribution, storage and loads. All the individual software platforms can complete a particular set task on their own, but they all just concentrate on the power network simulation. But one of the limitations of these types of simulation platforms is that the communication network which is a critical component of a Smart Grid is not considered in these platforms. This is why co-simulation platforms were introduced.

2.2.2 Co-Simulation Platforms

Co-simulation also known as co-operative simulation is a simulation approach that allows individual components to be simulated by different simulation tools running concurrently and exchanging information in a combined manner. In [12] Godfrey et al. simulated a Smart Grid using NS2 and OpenDSS which is a power network simulator. In [13], Mallouhi et al. created a co-simulation testbed just for security analysis of SCADA system by utilizing PowerWorld simulator and OPNET. The co-operative approach typically needs simultaneously running separate electrical and communication network simulations at the same time. The collaboration between communications and power system models is usually limited to a fixed synchronization interval. Reliability is an issue regarding systems like this because mismatches occurs between the simulations. An improvement about this issue is to integrate one simulation component into the other. In [14] electric network is made into a component within OMNET++, a network simulator.

From the above discussion, we can see the properties of the real hardware testbed approach and the software simulation approach for cyber-physical analysis in Smart Grid.

The real hardware testbed approach attains high fidelity by including dedicated devices as part of the testbeds. The critical control programs, such as demand response algorithms, routing protocols etc. can be tested in real hardware testbeds and they could be directly migrated to the actual Smart Grid embedded devices. However, the problems with the real hardware testbed approach is the scalability and accessibility factors. The dedicated and specialized hardware are integral parts of the testbeds therefore they cannot be easily accessed and used by the public research community and they become difficult to scale when the test case becomes quite large.

The software simulation approach, on the other hand, achieves better scalability and can be easily accessed and distributed. The software simulation tools can not duplicate the execution environment which is important, it can only duplicate behaviors of the Smart Grid system but not the execution environment. Therefore, the critical control programs of Smart Grid applications either cannot be tested or can be tested but cannot be migrated to physical Smart Grid devices directly.
CHAPTER THREE
Smart Grid Network Security and Simulator

3.1 Features of Smart Grid Networks

The smart grid network is expected to share similar architecture with the already existing Internet. However, there are important differences between them:

1. Latency requirements: The internet is created with the purpose of providing data services to the users like surfing or data sharing supported with high speed data rate. However, in smart grid this is quite not the case. Smart grid networks are intended for reliable, secure and real-time communications which are supported with low latency.

2. Communication model: In traditional power grids, the typical model for communication is one-way where electronic devices report their readings to the control center. But in smart grid, communication is bi-directional and real-time.

3. Data size and flow: Internet has generally bursty type communications however smart grid is expected to be bulky [15] and has periodic data communications because of the big size of the network and real time communication and monitoring requirements [1].

3.2 Smart Grid Network Simulator: The Score Application

In this project, we used SCORE for our simulations, an open research emulation environment for Smart Grid. SCORE is built upon CORE [16], an open source communication network emulator from the Naval Research Laboratory. Integrating CORE’s communication features with power module, SCORE differentiates itself from many existing approaches by enabling large scale Smart Grid applications using general purpose PCs while with little or no code migration problems. SCORE differentiates itself from the rest with the following specific features[23]:

- Firstly, software emulation in SCORE achieves high fidelity by replicating the execution environment so that the programs running in the emulation platform can be directly ported to the embedded devices as firmware.
- Secondly, SCORE enables distributed emulation feature in order for very large scale test cases to be supported.
Finally, SCORE supports dynamic connection and disconnection between multiple Smart Grid emulation instances in real time.

The significance of this feature is when users from multiple parties in different locations want to conduct the integration testing together, but want to preserve the privacy of power and communication networks configurations, this feature would make it happen without requiring explicit synchronization from all parties. The design of SCORE takes advantage of CORE’s structure. Figure 1 provides an abstract overview of SCORE’s architecture and the integration approach. As shown, SCORE consists of GUI, Service Layer, Communication Module and Power Module.

3.2.1 Graphics User Interface (GUI)

The SCORE GUI is built using Tcl/Tk. The Tk toolkit provides almost sufficient widgets for all the X window system interface needs. The Tcl/Tk GUI provides an easily drag-and-draw canvas with various Smart Grid devices (Host, Solar Panel, Wind Turbine, Power Plant etc.), which can be placed and connected to each other with communication links or power lines. Also, the communication interfaces, power interfaces and energy model parameters of each node can be self-configured.

During the execution, a terminal is popped out when double clicking any selected node. Users can navigate the local file system or execute bash script through the interactive shell window. Distributed emulation can be conducted by assigning a selection of nodes to another emulation server in GUI. The message broker in Service Layer is used to forward messages from the GUI to the appropriate emulation server.
3.2.2 Service Layer

The Service Layer consists of python frameworks that are used for creating sessions, instantiating the virtual nodes, communication and power interfaces, communication links and power lines, in regards to the GUI input. The start-up daemon in service layer cooperates with GUI using a TCP socket-based API such that the emulation can run on a different machine with the GUI or even without a GUI. Pre-defined energy models and communication protocols, which are usually daemonized in the Linux operating system of the emulation server, are all wrapped as Smart Grid services in this layer. These communication and energy services can all be employed to develop various Smart Grid Applications. Users are also allowed to add their own customized services to SCORE by providing their own implementations.
3.2.3 Light Weighted Virtualization

The emulation features of SCORE are executed using a Linux namespace technique, which is the light weighted paravirtualization technique supported by mainstream Linux kernel. It is different from the normal virtual machines techniques like VMware or Virtual Box. Each emulated virtual node in SCORE has its separated copy of network interface, protocol stack and process control group. All other resources like the operating system and local file system are all shared by the virtual nodes. The light-weighted virtualization feature is the basis of SCORE scalability ability. Furthermore, from the evaluation of the codes running inside the virtual node, each emulated device is given the impression of just another piece of hardware platform controlled by the Linux OS. This equips SCORE with the property of portability in order for the emulated node to be able to execute unmodified Smart Grid application codes running inside a real physical Linux-running hardware devices, and vice versa.

3.2.4 Communication Module

The communication module in SCORE leverages the across-the-board support of various wired and wireless communication network models and protocols from CORE. Each emulated device has its own instance of operating system implemented TCP/IP stack from the perspective of Open Systems Interconnection (OSI) model. This makes SCORE have the high fidelity emulation of network layer and above. Statistical network effects such as bandwidth, bit error rate, loss rate, etc. can also be configured and applied. In addition, the virtualized Ethernet interface can be easily mapped to a physical Ethernet interface on the emulation host so that all traffic going through the physical port would be transmitted to the emulation environment. Thus, allowing real time communication between the external physical networks and the virtual nodes inside a running emulation.

By using the virtualized interfaces on each emulated host, the communication network that is emulated on different hosts can then be directly connected with each other in run time, which enables the dynamic emulation of the communication networks. This feature is used to enable the interactions and synchronization between the communication module and the power module. The concept is that the power module is running on a host physically in the same network with the communication emulation host so that the power module can obtain and react to the queued-up messages sent by all the emulated virtual node in real time.
3.2.5 Power Module

The power module in SCORE emulates the power flows analysis within Smart Grid and also gives implementations of pre-defined energy models. The power module gathers initial power network topology, energy model configuration information and the dynamic connection/disconnection request from service layer to create the power network model. The power network module of SCORE is underlined by the following qualities:

- SCORE accepts incremental model updating in computation to respond more efficiently to the system status changes.

- As size of power network increases, distributed computation for power network becomes a requirement for an efficient Smart Grid emulation. Therefore, SCORE highlights itself in scalability by enabling the user to conduct the emulation in a distributed way when a single PC cannot provide enough computation capabilities. The power network model is split into several subdomains and each subdomains is computed and updated separately in parallel. With appropriate synchronizing among the different computing and updating processes, the merged result of the power flow in Smart Grid is compact without any loss of precision when compared with centralized computation.

- SCORE allows dynamic connections and disconnections of multiple Smart Grid instances running on different hosts by only using the interfaces between each power network. The importance of doing this is in the case when each user is unwilling to reveal their own Smart Grid topology details to another user, they can still conduct the combined emulation with each other to see the impact of external networks on their own network.

3.3 Smart Grid Security Requirements and Objectives

There are different factors to consider when discussing cyber attacks in smart grid systems. These factors include integration of bi-directional communication networks, incentives to attackers, socioeconomic impact of the blackouts, etc. Basically, the attack risk in the smart grid system relies on three factors as shown in Fig.2.

Formally, the risk can be defined as [17]:

\[ \text{Risk} = \text{Assets} \times \text{Vulnerabilities} \times \text{Threats}, \]
Fig. 2. Evaluating the risks in smart grid systems\cite{17}.

Assets are the smart grid devices (such as smart meters, renewable energy devices, data, network devices, etc.). Vulnerabilities allow an attacker to reduce a system's information assurance, and Threats may lead to potential attacks coming from outside or inside of the smart grid systems which are associated with the exploitation of a vulnerability. The risk is the probability that a threat agent will exploit a vulnerability and the impact if the threat is carried out. The ‘Risk’ in the above equation can be minimized or made zero if one of the quantities on the right side is minimized or made zero. It is therefore important to note that assets in smart grid systems cannot be zero and also threats cannot be made zero because they are originating from unknown places or attackers. Thus, the main aim and focus will be to minimize the vulnerabilities in the smart grid to minimize the overall ‘Risk’. Smart grid security objectives should be to comply with policies while ensuring information Confidentiality, Integrity and Availability, also known as the CIA triad. The CIA triad \cite{18} which is the fundamental principle of security is a model designed to guide policies for information security in smart grid systems. It is shown in Fig. 3.

Fig. 3. The CIA triad for smart grid security systems\cite{18}.
Confidentiality in the smart grid systems is needed to make sure that access to information is restricted to only authorized people and it is designed to prevent unauthorized access. Confidentiality is one of the key components of privacy. In smart grid systems, privacy is one of the most important concerns to customers. This is because of the various home appliances which are connected to power grids for real-time bi-directional data communication and electricity flow and if this information falls into the wrong hands, they can be used to keep track of the life style of the people, what appliances they use, whether the people are currently at home, etc. and misuse this information.

Integrity of information in smart grid is needed to ensure the accuracy and reliability of data. The information should not be altered in any form or undetected manner. This feature supports the smart grid in providing strong real-time monitoring capabilities.

Availability in the smart grid simply means that the information must be available to authorized parties at all times when it is needed and where ever it is needed without any security compromise. Power systems are to be available 100% of the time, therefore preventing attacker from implementing a blackout using denial-of-service is crucial. Additionally, Authenticity also plays a very important part in a smart grid system because it is essential to make sure that identities of both parties involved in communication are genuine.

In addition to the CIA triad, other specific security requirements for the smart grid recommended by NIST are outlined below [17], [18]:

1) Self-healing and Resilience Operations in the Smart Grid: In smart grid systems, the communication network is open as smart grid assets are distributed over large geographical area. Therefore, it is difficult to ensure that every single device in the smart grid is invulnerable to cyber-attacks. Because of these, it is advisable for the smart grid network to have some self-healing capability against cyber attacks. A network administrator must continually perform some sort of profiling and estimating to monitor the data flow and perform power flow status to detect any abnormal incidents that will be a product of cyber attacks. Having resilient data communication is very important to achieve availability of data communication for power system operations.

2) Authentication and Access Control: Because we have millions of home appliances connected in a smart grid, we need the authentication process to verify the identity of each device or user in order to protect smart grid systems from unauthorized access. Likewise, access control is used in smart grid to ensure that resources in the grid are accessed only by the authorized users.
3) Communication Efficiency and Security: In order to support real time monitoring, the smart grid communication needs to be efficient and highly secure together with the ability to use self healing cyber defense solutions to protect from any security attacks. Trade-offs between these two parameters should be considered in smart grids.

3.4 Smart Grid Attacks

The Three categories of smart grid cyber-attacks that we will discuss in this project are listed as follows:

1. Physical Layer Attacks,
2. Data Injection and Replay Attacks, and
3. Network-based Attacks.

3.4.1 Physical Layer Attacks.

There are several forms of physical layer attacks and a detailed analysis of some of the attacks and their countermeasures are given below [19]:

A. Eavesdropping

Wireless signals are transmitted in the air which is an open space and it is therefore susceptible to eavesdropping by an attacker. Sensitive information from a smart appliance can easily be observed, and compromised through such an attack. Eavesdroppers are readily available and affordable in today’s market which encourages such attacks. One way to protect against such attack is to use data encryption so as to protect some sensitive information from falling into the hands of an enemy. However, if a certain pattern is illustrated by the transmitted data, a smart hacker may use this pattern to create a way to decipher the messages transferred. For example, if everyone in a particular house is out for vacation, the electricity usage will drop. If the smart meter is instructed to communicate with the data concentrator unit if the length of the message to be transmitted is directly proportional to energy consumption, then a pattern of activity of the house can be generated by an attacker.

B. Jamming

The main aim of this type of attack is to disturb the wireless medium by jamming it with noise signals so that the smart meters can’t communicate with the utility provider. Such attacks can be either proactive jamming or reactive jamming.
The former is when the jammer emits noise signals continuously to completely block the wireless channel, while the latter is the case where the jammer first eavesdrops on the radio channel and launches the attack only when signals are sensed on the channel. This attack gives a bad result and affects when a legitimate smart meter tries to initiate a real connection. The channel maybe tagged busy for any carrier sensing done by the legitimate smart meter or it may even prevent it from receiving packets in general. It is quite difficult to differentiate between reactive jammer attacks that may be result from routine communication signals and from adversary-initiated signals.

C. Injecting Requests

The main goal of this attack is to disrupt the regular operations at the hardware level of devices in the smart grid. The attacker causes packet collision and it is similar to reactive jamming because it also blocks the communication channel. In injecting requests, the attacker sets the system in such a way that the channel prioritizes the attacker’s communication request while denying access to legitimate devices in the smart grid.

D. Injection Attacks

This attack inserts formatted messages into the wireless network unlike the earlier two attacks that depend on false signals. This type of attack involves an attacker mimicking either a legitimate sender or a receiver to get unauthorized access to a wireless network. This attack is also very similar to the TCP-SYN flooding (denial of service) attack wherein, the target's resources are overwhelmed through processing of false messages received. Such an attack can be avoided through providing the suitable security mechanisms to ensure message authentication.

3.4.2 Data Injection and Replay Attacks

Another class of malicious attacks in the smart grid is the data injection and replay attack. False data injection attacks occur when falsified data is injected into the neighborhood area observed by the network operator. The attacks usually target the smart grid infrastructure, particularly measurement and monitoring sub-systems with the aim of manipulating meter so as to deceive the operation and control of the utility provider.

Message replay attacks happen when an attacker gains elevated privilege to smart meters and as a result can then inject control signals into the system. For this attack to take place, the attacker needs to first capture and analyze the data that is transmitted between devices and smart meters to gain the targets characteristics of power usage, and then try to fabricate and
inject false control signals into the system. The main purpose of the replay attack is to control energy by directing power to another location, and another aim is to cause physical damage to the system. A well-known example of such an attack is Stuxnet.

In [20] a scheme is proposed for detecting message replay attacks in the smart grid. The household devices in the smart grid are treated as linear time invariant systems, with the smart meter assigned the role of observing the household devices. The replay attack is defined simply as a modification to the control signal which is communicated by a consumer device to the smart meter.

### 3.4.3 Network-Based Attacks

The man-in-the-middle attack is a very common example of topology attacks of a Smart Grid. This attack happens when the hacker captures network data and meter data from remote terminal units, and then tweak part of these in other to format and forward the altered version to the control center. If the smart grid is missing data alerts, the attacker can successfully alter both network and meter data efficiently such that they are consistent with the “target” topology.

A fusion-based defense technique was proposed in [21] for identifying attacks in the smart grid based on feedback received from individual nodes in the network. Through the support of the necessary communication protocol, each node is required to communicate with a centralized fusion center to convey their individual observations. It is highlighted in the paper, that intentional attacks may be targeted to only a specific subset of nodes of the smart grid, and therefore feedback from all nodes is essential for accurately detecting these attacks. A game theoretic analysis is subsequently provided, wherein, the attacker is treated as one player and the defender as another. Based on the notion that the attacker will intend to compromise the most critical nodes, the defense strategy is to ensure that timely local observation by individual critical nodes, and subsequent communication of findings to the centralized fusion center, is essential.

In [22], the effects of Denial of Service (DoS) attacks against the load frequencies of smart grids was studied. Smart grid data measured by remote terminals was sent to centralized control centers. If the communication channel between these sensors and the control center is compromised from delivering messages to the destination, the DoS attack can significantly affect the smart grid operations. The attacker can then launch such an attack on the communication channel by jamming the channel through injecting a large numbers of packets.
CHAPTER FOUR

SMART GRID SECURITY TESTBED

4.1 Testbed Requirements

The following requirements are necessary when configuring a testbed and they were implemented in our system:

R1 Modeling of Smart Appliance: This is considered to be a functional requirement. For the home area network emulation, the test bed should implement appliance emulation. There is a need for modelling smart appliances to be able to test the functionality of such devices, not only for technical reasons such as security assessment purposes, but also for operational reasons, such as to ensure that smart appliances are able to respond to price and demand signals to ensure that smart grid objectives and characteristics are fulfilled. Our system models different appliances in the grid.

R2 Hardware Integration: The test bed is expected to enable actual hardware integration or at least provide an interface to be able to integrate with actual hardware. This requirement is considered important as it will provide a realistic implementation of the test bed. Using physical hardware within the test bed will enable the evaluation and testing of real time characteristics. This also enables hardware testing without the need to set up and manage a hardware test bed environment.

R3 IP-based communication: To enable distributed use and remote access, IP based communication should be used between all major nodes of the test bed. This is an essential requirement not only for correct emulation of the current generation of smart grid network but also to enable distributed emulation, i.e. components of the test bed may be implemented and shared from geographically diverse networks to enable better utilization of resources. This would enable not only a collaborative test bed development and utilization but also foster innovation. IP based communication is inherent in smart grid test bed, as a majority of network communication paths in actual smart grids are based on IP networks. Thus smart grid networks are given greater flexibility, but also introduced to higher vulnerabilities that exist in IP based networks. Using IP based communications in smart grid test beds would also enable the rapid prototyping and assessment of IP based attacks and vulnerabilities.

R4 Graphical User Interface: A graphical user interface is another requirement that may be useful for an implemented test bed. A GUI will enable greater ease of use of the test bed. This will encourage greater participation, as well as provide an effective means to interact with the test bed for the design, development, and execution of test scenarios.
4.2 Testbed Configuration and Setup

The testbed for simulating the cyber security attack was implemented on a Linux-based host running a virtual machine using Oracle VirtualBox. The installation details for the Oracle VirtualBox are available in the Appendix. One of the primary areas of focus during the setup of our testbed was to employ open source and freely available software tools.

The testbed consists of two Linux-based virtual machines, one is dedicated to the attacker while the second system serves as the target machine. The attacker system runs the Kali Linux which is a Debian-derived Linux distribution designed for digital forensics and penetration testing. The Target machine on the other hand runs the Ubuntu Linux which is a Debian-based Linux operating system for personal computers, smartphones and network servers. The Installation steps for both Linux machines can be found in the Appendix section.

On the target machine, we created a Smart Grid Power Distribution network using Score. Score was installed in the Ubuntu-based target machine to also show the result of distributed emulation and dynamic connection/disconnection involved in a smart grid. The target system named “target@ubuntutarget” was given a memory of 1.2GB, using processor Intel Core i5-4200U CPU @ 1.60GHz and has a 64-bit OS Architecture running on the Ubuntu 15.10 version.

We generated a smart grid network with one power plant and three houses. Each house is connected with the power network through an intelligent power switch, which serves as the energy control center for the house. Each power switch within each house is connected to four different types of nodes:

- Loads (represented by washer)
- Power Storage (represented by battery)
- Renewable resources (represented by wind turbine and solar panel)

Fig. 4 Running Score
In order to begin setting up the smart grid network on the target machine, we have to start the SCORE services. This automatically runs the score-daemon program and starts the service. Afterwards, we go into the directory where score is installed and launch the SCORE GUI. All these steps can be seen in Fig. 4 above.

The smart grid network was initially set up with 16 nodes. Each node’s new interfaces are automatically assigned IP addresses within the range 10.0.0.0/8 and also support IPv6 address. Virtual networks generally require some form of routing in order to work, for example to populate routing packets from one subnet to another. Therefore, Score builds OSPF routing protocol configurations by default. The OSPF protocol is made available from the Quagga open source routing suite.

Fig. 5 Smart Grid Power Distribution Network set-up on Target Machine
Figure 5 above shows all the nodes in the smart grid network. Each node is connected through a wired network, which is created using the Link Tool, which allows drawing links between nodes. This automatically draws a green line representing an Ethernet link and creates new interfaces on network-layer nodes. Double clicking on each link will invoke the link configuration dialog box where we can change the Bandwidth, Delay, Loss and duplicate rate parameters for that link. The following services zebra, OSPFv2, OSPFv3, vtysh, and IPForward for IGP link-state routing are running on all nodes.

![Image of network diagram]

Fig.6. Screenshot showing the Throughput on each link when running.

The rate of all successful messages delivered over the communication link also known as the throughput can be displayed on each link as seen in Fig.6. The Widget Throughput in score displays the throughput measure in kilobit-per-second on each link.
To test and verify the communication between all nodes in the smart grid, we use the `ping -R` command. We issue the command from node 4 which is a power switch in one of the houses to contact node-20 which is the power plant supplying energy to the grid. The power switch has the address 10.0.15.2 and the node-4 address is 10.0.1.1. We Double click on a node to pop out a Linux terminal window, just like accessing a real Linux device. Fig.7 shows the result of the `route` command from the terminal of the power switch in the house and also the result of the `ping -R` command.

![Fig.7 Showing the connectivity test between all nodes in the system](image)
4.3 Attacks on the Smart Grid Testbed

The main aim of this project is to enable the simulation of attacks from the Kali Linux platform to the target Ubuntu system which is running the smart grid power distribution network. A GRE tunnel was created to enable the connection between the attacker system and the target system. The tunnel was on a router connected to node-3 in the smart grid on the target system. This GRE tunnel connection enables the outside network connection of the attacker to have full connectivity to all the components in the smart grid.

Before we start with the attack, we use a network discovery tool called Nmap to explore the network we are about to target. This tool is preinstalled on Kali-Linux and is useful in gathering important information about a network like IP addresses, host details, services, port details and much more. This tool will enable us to map out the network and understand the network topology. Fig. 8 shows a view of the topology of the target system using the fisheye view in Nmap.

Fig.8 Fisheye Topology View of the Smart Grid Network
Here is the end of the *Nmap* output in Fig. 9. It shows the total number of hosts up, the services running on each device and a lot of other useful information.

As discussed earlier, there are many attacks that can affect the smart grid system. As an example and case study, in this project we are going to launch a denial of service (DoS) attack against the power grid. One of the worst attacks against a smart grid is the DoS attack, as a successful attack can severely limit or prevent access to important devices or services. We launch the DoS on the power plant which supplies energy to the grid. By doing this, the smart grid is compromised by eventually shutting down the link that provides the energy and broadcasts the real time energy prices to all the intelligent power switches in the grid.

---

![Fig. 9 Nmap Output](image1.png)

![Fig. 10 DoS command using hping3](image2.png)
Our DoS attack was performed using a free packet generator and analyzer tool for the TCP/IP protocol called *hping3* tool. Fig. 10 shows the line of command used to perform the DoS attack. The *hping3* tool is pre-installed on Kali Linux like many other tools. The syntax of the command is explained as follows:

- `-c 100000` = Number of packets to send.
- `-d 120` = Size of each packet that was sent to target machine.
- `-S` = Send SYN packets only.
- `-w 64` = TCP window size.
- `-p 2601` = Destination port (2601 being the TCP port analyzed from *Nmap*).
- `--flood` = Send packets as fast as possible, without taking care to show incoming replies. Flood mode.

Fig. 11 Screenshot of the DoS attack
Following the DoS attack initiated from the attacker's Kali Linux box, we can see from the screenshot in Fig. 11 the effect of the attack on each link leading to the target with address 10.0.15.2. The Throughput on these links jumped from 8.7kpbs in Fig. 6 to about 8700kbps during the attack. A SYN Flood type of DoS was used in this attack. TCP SYN flood also known as Synflood is a type of DoS attack that exploits part of the normal TCP three-way handshake to consume resources on the targeted system and render it unresponsive. In this case it renders the powerplant unresponsive. The normal TCP three-way handshake follows this step:

1. Client requests connection by sending SYN (synchronize) message to the server.
2. Server acknowledges by sending SYN-ACK (synchronize-acknowledge) message back to the client.
3. Client responds with an ACK (acknowledge) message, and the connection is established.

In the SYN flood attack, the attacker system sends repeated SYN packets using hping3 to known port 2601 on the target system. The target, unaware of the attack, receives multiple, apparently legitimate requests to establish communication. It responds to each attempt with a SYN-ACK packet from the open port.

```
root@4:/tmp/pycore.44506/n4.conf# ping 10.0.15.2
PING 10.0.15.2 (10.0.15.2) 56(84) bytes of data.
64 bytes from 10.0.15.2: icmp_seq=1 ttl=63 time=1.80 ms
64 bytes from 10.0.15.2: icmp_seq=2 ttl=63 time=1.78 ms
64 bytes from 10.0.15.2: icmp_seq=3 ttl=63 time=1.95 ms
64 bytes from 10.0.15.2: icmp_seq=4 ttl=63 time=1.85 ms
64 bytes from 10.0.15.2: icmp_seq=5 ttl=63 time=1.93 ms
64 bytes from 10.0.15.2: icmp_seq=6 ttl=63 time=1.70 ms
64 bytes from 10.0.15.2: icmp_seq=7 ttl=63 time=1.77 ms
64 bytes from 10.0.15.2: icmp_seq=8 ttl=63 time=1.81 ms
64 bytes from 10.0.15.2: icmp_seq=9 ttl=63 time=1.88 ms
64 bytes from 10.0.15.2: icmp_seq=10 ttl=63 time=1.81 ms
64 bytes from 10.0.15.2: icmp_seq=11 ttl=63 time=1.96 ms
64 bytes from 10.0.15.2: icmp_seq=12 ttl=63 time=1.84 ms
64 bytes from 10.0.15.2: icmp_seq=13 ttl=63 time=1.80 ms
64 bytes from 10.0.15.2: icmp_seq=14 ttl=63 time=2.00 ms
From 10.0.1.2 icmp_seq=50 Destination Host Unreachable
From 10.0.1.2 icmp_seq=51 Destination Host Unreachable
From 10.0.1.2 icmp_seq=52 Destination Host Unreachable
From 10.0.1.2 icmp_seq=53 Destination Host Unreachable
From 10.0.1.2 icmp_seq=54 Destination Host Unreachable
From 10.0.1.2 icmp_seq=55 Destination Host Unreachable
From 10.0.1.2 icmp_seq=56 Destination Host Unreachable
From 10.0.1.2 icmp_seq=57 Destination Host Unreachable
From 10.0.1.2 icmp_seq=58 Destination Host Unreachable
From 10.0.1.2 icmp_seq=59 Destination Host Unreachable
From 10.0.1.2 icmp_seq=60 Destination Host Unreachable
From 10.0.1.2 icmp_seq=61 Destination Host Unreachable
From 10.0.1.2 icmp_seq=62 Destination Host Unreachable
From 10.0.1.2 icmp_seq=63 Destination Host Unreachable
From 10.0.1.2 icmp_seq=64 Destination Host Unreachable
From 10.0.1.2 icmp_seq=65 Destination Host Unreachable
From 10.0.1.2 icmp_seq=66 Destination Host Unreachable
From 10.0.1.2 icmp_seq=67 Destination Host Unreachable
From 10.0.1.2 icmp_seq=68 Destination Host Unreachable
From 10.0.1.2 icmp_seq=69 Destination Host Unreachable
From 10.0.1.2 icmp_seq=70 Destination Host Unreachable
From 10.0.1.2 icmp_seq=71 Destination Host Unreachable
From 10.0.1.2 icmp_seq=72 Destination Host Unreachable
From 10.0.1.2 icmp_seq=73 Destination Host Unreachable
From 10.0.1.2 icmp_seq=74 Destination Host Unreachable
From 10.0.1.2 icmp_seq=75 Destination Host Unreachable
From 10.0.1.2 icmp_seq=76 Destination Host Unreachable
From 10.0.1.2 icmp_seq=77 Destination Host Unreachable
From 10.0.1.2 icmp_seq=78 Destination Host Unreachable
From 10.0.1.2 icmp_seq=79 Destination Host Unreachable
From 10.0.1.2 icmp_seq=80 Destination Host Unreachable
```

Fig.12. A ping from a node to the target during the DoS attack
The attacker does not send the expected ACK while the powerplant under attack still waits for acknowledgement of its SYN-ACK packet for some time. During this time, the powerplant cannot close down the connection by sending an RST packet, and the connection stays open. Before the connection can time out, another SYN packet will arrive from the attacker. This leaves an increasingly large number of connections half-open. Eventually, as the target’s connection overflow tables fill, service to legitimate nodes in the smart grid distribution will be denied, leading to the target becoming unreachable. A verification of our successful attack can be seen in Fig.12. Here we launch a ping from node-4 with address 10.0.1.2 to the powerplant with address 10.0.1.15 right before and after the DoS attack.
CHAPTER FIVE

CONCLUSION

In this report, we discussed about smart grid and various attacks affecting it. We also designed, implemented and attacked a simulated smart grid power system using a form of Denial of Service. A smart grid infrastructure attack does not affect the consumers alone, rather, the utility providers' business as well.

Extensive research work is still needed to ensure that the smart grid is highly secure against the adversarial threat, without affecting the consumer confidence in the utility provider, and without significantly inconveniencing the consumers through deployment of strong security controls.

The testbed will provide a platform for researchers to execute various attack scenarios and study their impact on smart grid networks. This would allow designing adequate protection for smart grid infrastructure networks.

One future direction would be integrating SCORE with real hardware testbed to create a uniform cyber-physical analysis platform.
APPENDIX A

INSTRUCTIONS

The instructions provide a step–by–step guide to what commands where executed. This includes instructions for the installation, configuration and execution of components of the test bed implementation.

A.1 SCORE

SCORE is built based on CORE, an open source communication network emulator from Naval Research Laboratory. TCL/TK GUI and the communication network component are originated from IMUNES project from the University of Zagreb. The Linux virtualization and the python frameworks for Linux namespace and communication network have developed by Boeing Research and Technology’s Network Technology research group since 2004.

A.1.1

1. tar xvzf SCORE1.0.tar.gz
2. cd SCORE1.0
3. make
4. sudo make install

A.2 Oracle Virtualbox

VirtualBox is a cross-platform virtualization application. It is deceptively simple yet also very powerful. It can run everywhere from small embedded systems or desktop class machines all the way up to datacenter deployments and even Cloud environments. You can install and run as many virtual machines as you like – the only practical limits are disk space and memory.

A.2.1

1. sudo apt-get install dkms
2. sudo dpkg -i virtualbox-5.0_5.0.16_Ubuntu_raring_i386.deb
3. sudo ./VirtualBox.run install
4. ./VirtualBox.run --keep --noexec
5. sudo mkdir /opt/VirtualBox
6. sudo tar jxf ./install/VirtualBox.tar.bz2 -C /opt/VirtualBox
7. make
8. `sudo make install`
9. `make install`
10. `cp /opt/VirtualBox/vboxdrv.sh /sbin/rcvboxdrv`
11. `mkdir /etc/vbox`
12. `echo INSTALL_DIR=/opt/VirtualBox > /etc/vbox/vbox.cfg`
    and, for convenience, create the following symbolic links:
13. `ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VirtualBox`
14. `ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxManage`
15. `ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxHeadless`
16. `ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxSDL`

### A.3 KALI LINUX

Kali Linux is a Debian-derived Linux distribution designed for digital forensics and penetration testing. Kali Linux is preinstalled with over 300 penetration-testing programs. Kali Linux can run natively when installed on a computer’s hard disk, can be booted from a live CD or live USB, or it can run within a virtual machine. It is a supported platform of the Metasploit Project’s Metasploit Framework, a tool for developing and executing security exploits.

#### A3.1

2. Burn The Kali Linux ISO to DVD or Image Kali Linux Live to USB.
3. Ensure that your computer is set to boot from CD / USB in your BIOS.
4. Boot your system with your chosen installation medium and follow on screen instructions.

### A.4 UBUNTU LINUX

Ubuntu is a Debian-based Linux operating system and distribution for personal computers, smartphones and network servers. It uses Unity as its default user interface. It is based on free software and named after the Southern African philosophy of *ubuntu* (literally, "human-ness"), which often is translated as "humanity towards others".

#### A 4.1

1. Download Ubuntu from the official download page ([http://www.ubuntu.com/download](http://www.ubuntu.com/download))
2. Burn Ubuntu Linux ISO to DVD or Image Ubuntu Linux Live to USB.
3. Ensure that your computer is set to boot from CD / USB in your BIOS.
4. Boot your system with your chosen installation medium and follow on screen instructions.
REFERENCES

