Development of smart grid testbed with low-cost hardware and software for cybersecurity research and education

ANNOR-ASANTE, Michael and PRANGGONO, Bernardi
<http://orcid.org/0000-0002-2992-697X>

Wireless Pers Commun (2018) 101:1357–1377
https://doi.org/10.1007/s11277-018-5766-6

Development of Smart Grid Testbed with Low-Cost Hardware and Software for Cybersecurity Research and Education

Michael Annor-Asante[1] · **Bernardi Pranggono**[1]

Published online: 23 April 2018

Abstract Smart Grid, also known as the next generation of the power grid, is considered as a power infrastructure with advanced information and communication technologies (ICT) that will enhance the efficiency and reliability of power systems. For the essential benefits that come with Smart Grid, there are also security risks due to the complexity of advanced ICT utilized in the architecture of Smart Grid to interconnect a huge number of devices and subsystems. Cybersecurity is one of the emerging major threats in Smart Grid that needs to be considered as the attack surface increased. To prevent cyber-attacks, new techniques and methods need to be evaluated in a real-world environment or in a testbed. However, the costs for setting-up Smart Grid testbed is extensive. In this article, we focused on the development of a Smart Grid testbed with a low-cost hardware and software for cybersecurity research and education. As a case study, we evaluated the testbed with most common cyber-attack such as denial of service attack. In addition, the testbed is a useful resource for cybersecurity research and education on different aspects of SCADA systems such as protocol implementation, and PLC programming.

Keywords Cybersecurity · Denial of service · Intrusion detection system · Modbus · Testbed · SCADA · Smart Grid

✉ Bernardi Pranggono
b.pranggono@shu.ac.uk

Michael Annor-Asante
mikeasante100@hotmail.com

[1] Department of Engineering and Mathematics, Sheffield Hallam University, Sheffield, UK

1 Introduction

Electricity is very important in our daily life as it is used for the operation of many electronics devices from our homes to offices to hospitals and manufacturing industries. Due to its importance, it is crucial to maintain its availability as high as possible and to be used in the most efficient manner as its resources are limited.

Some developing countries face the problem of not having the ability to generate and provide electric power on a continuous basis to their cities to produce goods and services and for daily activities due to poor use and management of resources. This problem has prompted for the research of a system with advanced computing technologies for the safe, efficient and reliable way of delivering electric power—the Smart Grid.

Smart Grid is an essential part of sustainable smart cities. Smart Grid is the next generation of power grid in which the management and distribution of electricity are performed in advanced two-way communication systems. The Smart Grid system is made up of subsystems that are interconnected to provide the flow of information and electricity in a two-way communications using information and communication technologies (ICT) for safer, cleaner and more reliable electricity [1]. The extensive use of cyber infrastructure presents serious consequences with respect to security of physical systems [2–4].

In the Smart Grid there are four subsystems: generation, transmission, distribution and advanced metering subsystems. These systems must work together in the best possible manner (smart) and be resilient to physical and cyber-attacks. Typically, a Smart Grid is a combination of [1]:

1. Smart infrastructure system: a modernized system of the conventional power grid.
2. Smart management system: management mechanisms deployed within the infrastructure for safe, clean and reliable electricity.
3. Smart protection system: security implementations to protect the system from physical and cyber-attacks to ensure that the security objectives of the system (Confidentiality, Integrity and Availability) are achieved.

Cybersecurity is one of the emerging major threats in Smart Grid that needs to be considered as the attack surface increased [3]. To prevent cyber-attacks, new techniques need to be evaluated in a real-world environment or in a testbed. However, the costs for setting-up Smart Grid laboratory and testbed is extensive. In fact, the average expenditure to build a Smart Grid laboratories are in the order of 2 million euro [5]. Such cost is a significant obstacle to build new laboratories for research and education purposes. In this article, we focused on the development of a Smart Grid testbed with a low-cost hardware and software for cybersecurity research and education. The design objective is such that it able to simulate accurately real-world cyber-attacks scenario using low-cost hardware and software that can be used in the classroom. In fact, our Arduino based testbed can be developed with the cost of less than 150 lb sterling (excluding the computers).

The main contributions of the article are:

- The development of real-world Smart Grid testbed using low-cost hardware and software.
- The development of SCADA intrusion detection system (IDS) and real-world cyber-attacks scenarios.

The remainder of the article is organized as follows. Section 2 provides background information on the Smart Grid with focus on its characteristics. Section 3 discusses the method implemented in developing Smart Grid testbed using low-cost hardware and

software. In Sect. 4, the simulation results from our case study experiments using distributed denial of service (DDoS) attack are discussed. Finally, a conclusion is highlighted in Sect. 5.

2 Background

2.1 Smart Grid

Over the last century, the traditional power grid has functioned with the involvement of centralized power plants that fed power over a one-way channel from the distributor to the consumer. There has been a burden on the traditional power grid with several technical and economic issues [6]. This problem has led to the development of the next generation of power system known as Smart Grid which integrates advanced ICT for more efficient and reliable power systems (Fig. 1).

Smart Grid is a modern power system infrastructure that has many advantages. Some of the characteristics of the Smart Grid are:

Self-Healing Smart Grid may redirect the flow of electricity from distribution points to homes and industries when there is an interruption in an electrical transmission path due to severe weather conditions, technical problems, etc. This is one of the functionalities and advantageous factors that can automatically adjust the flow of electricity to prevent power loss that could affect the production of goods and services.

Resilience to Attacks Due to its complexity that interconnects generation, transmission, distribution and advanced metering subsystems, it has a wider area of security vulnerabilities that needs to be addressed. The Smart Grid is designed to be resilient to physical

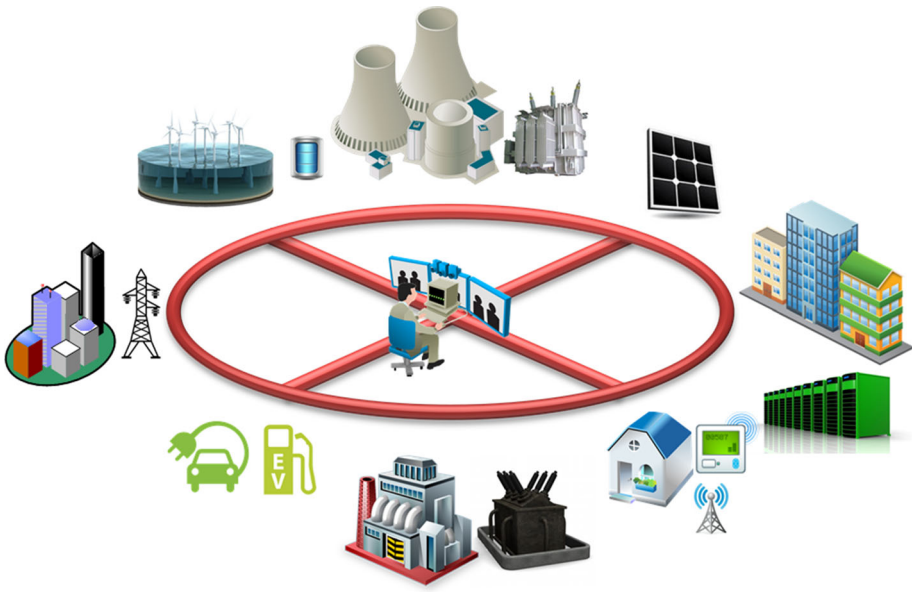


Fig. 1 Smart Grid

attacks, cyber-attacks, and natural disasters as it is one of national critical infrastructure that crucial for the economic welfare and security.

Increase in Power Quality Availability of electricity is not the only important thing to be considered but the quality in terms of maintaining a constant voltage is also equally important. Manufacturing industries rely on power quality for production therefore voltage variations could cause productivity losses which will in turn to cost increase. The Smart Grid increases the quality of power through effective transmission from power plants and distribution from step down transformers.

The security requirements in Smart Grid are: attack detection and resilience operations, secure and efficient communication protocols, identification, authentication and access control. The security of the Smart Grid is an important issue because a disruption will have significant societal and economic impacts [2–4].

The Smart Grid uses control system applications such as supervisory control and data acquisition (SCADA) integrated with various industrial communication protocols such as DNP3, Modbus, PROFIBUS, Zigbee, Profinet, EtherNet/IP, etc. to control and monitor various physical aspects of the Smart Grid's subsystems. A SCADA typically communicates with external devices via the SCADA communicator (protocol handlers such as Modbus, DNP3, etc.).

Generally, Smart Grid network architecture comprises of:

1. *Generation System Architecture* Corresponds to specific controller logic used to automate the generation process of power.
2. *Transmission System Architecture* Transmission mechanisms to the distribution points and to end-users.
3. *Distribution System Architecture* Distribution systems comprise of substations for energy conditioning, monitoring and automation. These substations communicate back to central SCADA systems in data centers for real-time monitoring, load balancing and energy management and to respond to power outages.
4. *Advanced Metering Architecture* This architecture consists of systems that are used not only to measure energy utilization but also to remotely connect or disconnect meters. The use of smart meters adds the capability of sending commands from a centralized system and receiving responses. It consists of three primary components: smart meters, a communication network, and an advanced metering infrastructure (AMI) server or head-end [7].

Some of the motives of cyber-attacks on the Smart Grid are: theft of information, denial of service (DoS) and manipulation of service (MoS). The attack tools are used to conduct various attack methods such as man-in-the-middle (MITM) attack, replay attack, human–machine interface (HMI) attack, NAN attack, jamming attack, DOS attack, bad data injection, etc. [8, 9].

2.2 Intrusion Detection System

Intrusion detection is a process of using a set of methods (detection system) to detect anomalous activity at the host and network level. Intrusion detection systems (IDS) are software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems [10]. IDS can be classified into two categories: network-based and host-based [11].

Host-based IDS (HIDS) is installed as a monitoring agent to protect a single host or endpoint and monitors activities on the host system. HIDS monitors the system and application files to detect any malicious activity and alert when something anomalous has happened. HIDS are reactive and proactive; it can sniff packets coming to a host and alert when a signature pattern is found [10].

Network-based IDS (NIDS) on the other hand, protects the whole network from attacks by monitoring traffic on the network. It captures every packet travelling across the network and analyses against a database of signatures. One of the most popular NIDS applications is Snort [12]. To prevent malicious activities in the network, NIDS typically uses two different detection methods: signature-based and anomaly-based detection.

Signature-based detection works more like anti-virus software and has a database of known attack signatures which need to be updated regularly. Snort is the de facto standard of signature-based NIDS. Snort has a decoder for normalizing packets, a detection engine for comparing packets with signatures, a logger for recording packet details and an alert system for triggering alarms. When a packet is matched with an intruder signature an alert is generated and the packet is logged to a database [12].

With anomaly-based detection, the NIDS starts with a known good state and use that to establish a baseline of normal activity. It compares the current state with the baseline and alerts when an anomaly is detected [11].

IDS supports active and passive response or a mixture of the two responses. With active response, automated actions take place when certain types of intrusions are detected. These actions are put in place to collect additional information on a suspected attack to try to resolve the detection of an attack and gather information to support the investigation. Another automated action is to halt an attack in progress and block access by reconfiguring routers and firewalls to block network ports, services and protocol that are being used by an attacker or terminating an attacker's connection by injecting TCP reset packets [10].

Passive responses include alarms and notifications and SNMP traps [10]. They only provide information to users so that certain actions or decisions can be made based on the information provide. Most of the IDSs rely on this type of response.

The response of IDS has four possibilities [10]:

1. True positive: real attack—IDS alert
2. True negative: no attack—no IDS alert
3. False positive: no attack—IDS alert
4. False negative: real attack—no IDS alert

2.3 Denial-of-Service (DoS) Attack

Denial-of-service is one of the most common cyber-attacks in the last decade. The objective of this attack is to disrupt a system from its normal operation by sending messages to the victim machine to crash, reboot or operate in an improper manner. Attackers exploit the vulnerabilities on the target systems or applications running in the systems and use that means to attack the systems. Attackers' main goal is to compromise the availability of services [2].

Typically, a DoS attack is achieved by sending a vast number of packets to consume key resources such as bandwidth, CPU, memory, etc. on target systems or network thereby making the victim spend its resources in handling or responding to the attack traffic. DoS attacks are mostly launched from single point powerful machines with fast processors and a lot of network bandwidth.

A DDoS attack is a derivative of DoS attack that uses many hosts to attack. These hosts are known as 'bots' or 'zombies'. A collection of bots (botnet) operates under the control of a 'master' to launch DDoS attacks by sending some traffic from different sources to a target to overwhelm or exhaust the target's resources [2, 13].

In reality, DDoS attack is easily performed by open-source DDoS attack tools such as Bonesi, Low Orbit Ion Cannon (LOIC) [14] or High Orbit Ion Cannon (HOIC) [15].

3 Methodologies

In this article, the approach is to develop a testbed of the Smart Grid infrastructure using combination of various low-cost hardware and software such as the Arduino microcontroller [16] (PLC), XBee radio modules (wireless data acquisition modules), Winlog Lite [17] (SCADA), Bonesi [18] (botnet simulator), Snort [12] (IDS), Suricata [19] (IPS) and exploit the system with a cyber-attack application.

3.1 System Configuration

The overview of the system is illustrated in Fig. 2. Zigbee [20] is adopted as the main communication protocol between subsystems. An Arduino microcontroller is connected to a personal computer using a serial cable. One of the XBee module is connected to the Arduino as an interface for data transmission to the SCADA server and for sending commands from the server via the serial communication port of the computer. The SCADA server uses Modbus TCP/IP [21] as a communication protocol to send commands and receive data from the field devices. Modbus is an application layer protocol used for communication in a master/slave architecture in industrial automation. It was developed in 1978 by Modicon (now Schneider Electric) for use with PLCs [22]. The protocol is similar

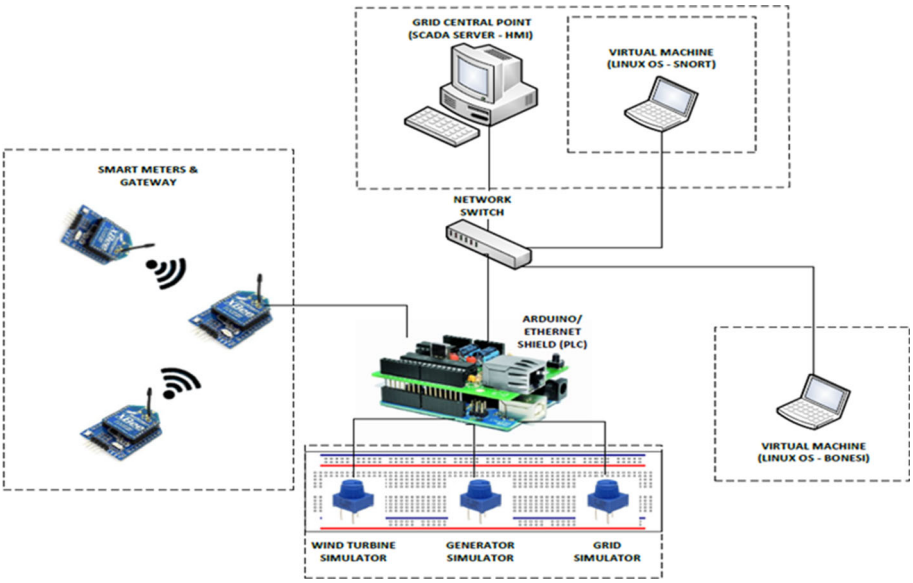


Fig. 2 System architecture

to HTTP protocol which consists of a request by a client and a response issued by a server. In the client–server paradigm of the Modbus protocol, the master is a client and the slave is a server because the master sends requests to the slave holding data in the Modbus memory block. The protocol was first implemented for asynchronous serial network communication (RS232/RS485) and has now been developed to use the TCP/IP stack for communication on port 502 (Modbus TCP/IP) [21].

XBee communication is used within the system. The Arduino will be connected to a personal computer using a serial cable. One of the XBee modules will be connected to the Arduino as an interface for data transmission to the SCADA server and for sending commands from the server via the serial communication port of the computer. The SCADA server will use Modbus on port 502 as a communication protocol to send commands and receive data from the field devices.

There are two communication paths in the system: one from the microcontroller to the computer (using SCADA protocol) and the other from the microcontroller to the XBee modules (using XBee protocol) and potentiometers. The XBee modules are configured as DigiMesh nodes to simplify the implementation of the Smart Grid. The SCADA is simulated by Winlog Lite. The Winlog Lite is used in conjunction with the Modbus TCP protocol to communicate with the devices, create process variables (tags) to store the data within the SCADA server and design the graphical user interface (dashboard) of the Smart Grid system. An Arduino microcontroller is programmed to read analog and digital values from the XBee modules and pass them to the SCADA server, pass on commands from the SCADA server to the XBee modules and external devices and to make intelligent decisions as expected of a Smart Grid system. To evaluate the proposed system, a DoS cyber-attack is implemented to study the behavior of the system and a preventive measure was also put in place.

3.2 System Architecture

As illustrated in Fig. 2, there is a remote Local Area Network (LAN) which consists of smart meters and a gateway or a communication mechanism for the transmission of data to the grid central point via a PLC (Arduino). There is also a power supply network which consists of the grid, a generator and a wind turbine. These power generation units send data to the grid central point via a PLC (Arduino).

The Grid network consists of the SCADA server as an HMI, IDS, a network switch and a PLC (Arduino). A cyber-attack (Bonesi) machine is also connected to a network switch to implement the cyber-attack (DoS) on the Smart Grid system.

The circuit diagram in Fig. 3 shows two XBee router radios acting as smart meters in remote locations. These radios are connected to a central XBee (coordinator) in a personal area network. There are two potentiometers connected to the remote radios to simulate power consumption from different homes. The XBees send the data wirelessly to the coordinator XBee for transmission to the SCADA server via the serial communication port on the Arduino.

There are other potentiometers connected directly to the Arduino as the Grid utility, generator and wind turbine (power generation units). The LEDs are indicators used to monitor various parts of the system and the dip switches are simply used to control the flow of power.

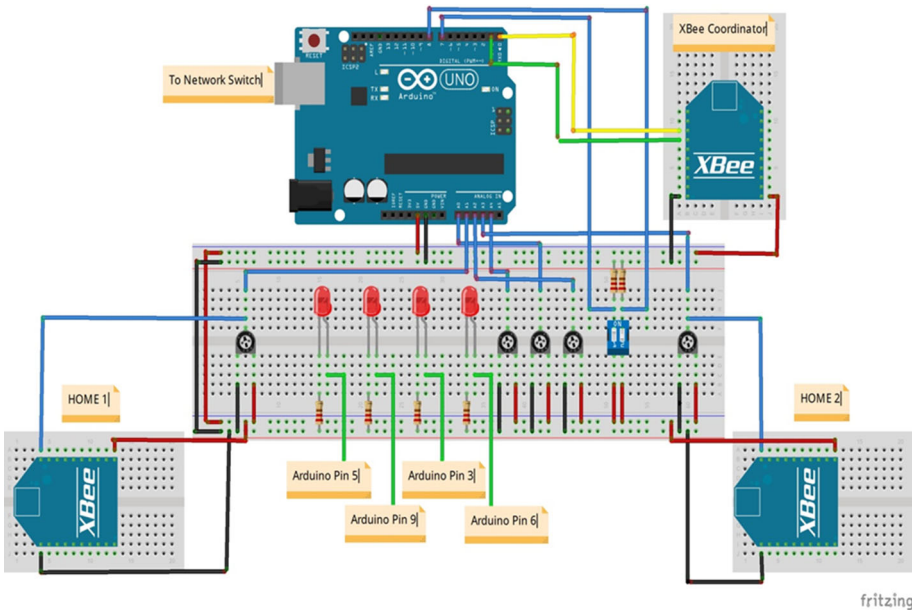


Fig. 3 Circuit diagram

3.3 SCADA System

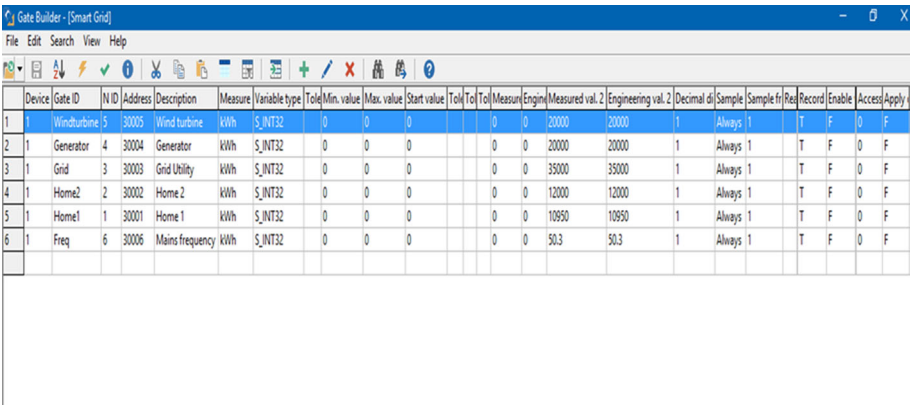
The SCADA application is configured by creating a Modbus device driver which is used to communicate with the Arduino (PLC). The Modbus configuration such as device ID, Modbus addresses in the SCADA application should be the same as PLC configuration. This ensures that data is transferred effectively from the PLC to the SCADA application. The Modbus protocol is much like the HTTP protocol which consists of an initiation of a request by a client and a response issued by a server. In the client–server paradigm of the Modbus protocol, the master is a client and the slave is a server because the master sends requests to the slave holding data in the Modbus memory block.

The SCADA application also contains database tags for storing data from PLC and displaying the data in different forms on a human–machine interface (HMI) (see Figs. 4, 5). The HMI is built with designing tools within the SCADA application for monitoring and controlling purposes.

3.4 PLC Configuration

The Arduino microcontroller is programmed to be used as a PLC by compiling the Modbus TCP functions in a library to be loaded on to the board. The PLC is used as a slave to respond to requests from the SCADA server. The serial peripheral interface library, Modbus library and Ethernet library are uploaded onto the microcontroller board to facilitate the implementation of the Modbus protocol and transfer of Modbus data.

An IP address is configured for the Ethernet shield on the Arduino board to enable data exchange with the SCADA server via Ethernet port. The IP address of the PLC is the slave address and the IP address of the SCADA server is the master address of the PLC. The addresses of the PLC and SCADA server are typically in the same subnet or in different



	Device	Gate ID	N ID	Address	Description	Measure	Variable type	Tole	Min. value	Max. value	Start value	Tole	Tol	Tol	Measure	Engine	Measured val. 2	Engineering val. 2	Decimal di	Sample	Sample fr	Rel	Record	Enable	Access	Apply
1	1	Windturbine	5	30005	Wind turbine	kWh	S_INT32		0	0	0				0	0	20000	20000	1	Always	1		T	F	0	F
2	1	Generator	4	30004	Generator	kWh	S_INT32		0	0	0				0	0	20000	20000	1	Always	1		T	F	0	F
3	1	Grid	3	30003	Grid Utility	kWh	S_INT32		0	0	0				0	0	35000	35000	1	Always	1		T	F	0	F
4	1	Home2	2	30002	Home 2	kWh	S_INT32		0	0	0				0	0	12000	12000	1	Always	1		T	F	0	F
5	1	Home1	1	30001	Home 1	kWh	S_INT32		0	0	0				0	0	10950	10950	1	Always	1		T	F	0	F
6	1	Freq	6	30006	Mains frequency	kWh	S_INT32		0	0	0				0	0	50.3	50.3	1	Always	1		T	F	0	F

Fig. 4 SCADA server Gate Builder window

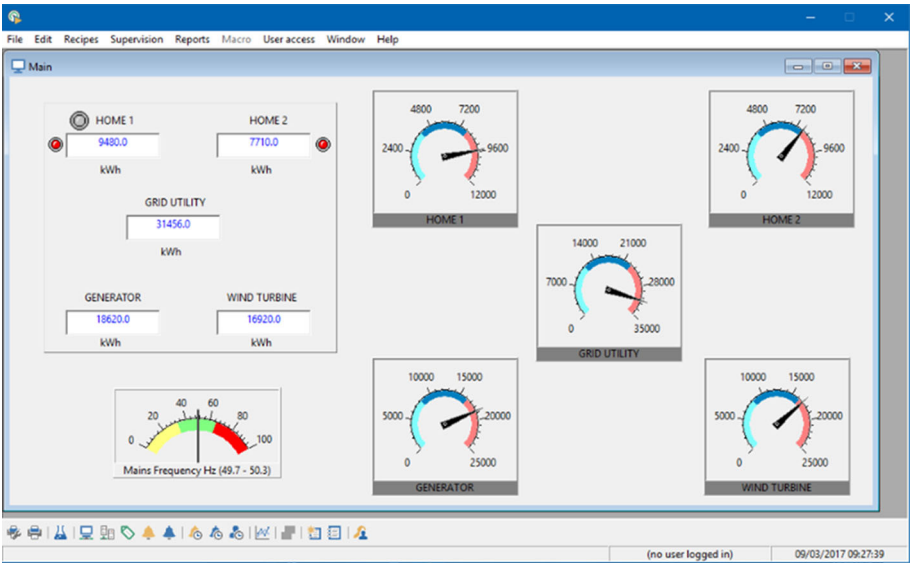


Fig. 5 SCADA server graphical interface (HMI)

subnets with routers between subnets. The simulation tools used for the Smart Grid infrastructure are connected to analog and digital inputs/outputs of the Arduino. The program code is written such that the Arduino reads data from the analog pins and writes the data to the holding registers of the Modbus addresses for transfer to the SCADA server on request.

To implement the basic functionality of Smart Grid, the program code reads the data coming from the XBees (home simulators) via the XBee coordinator which is connected to the serial TX and RX on the Arduino (see Fig. 6). A value of zero signifies a power cut, resulting in system notification and automatic restore of outage.

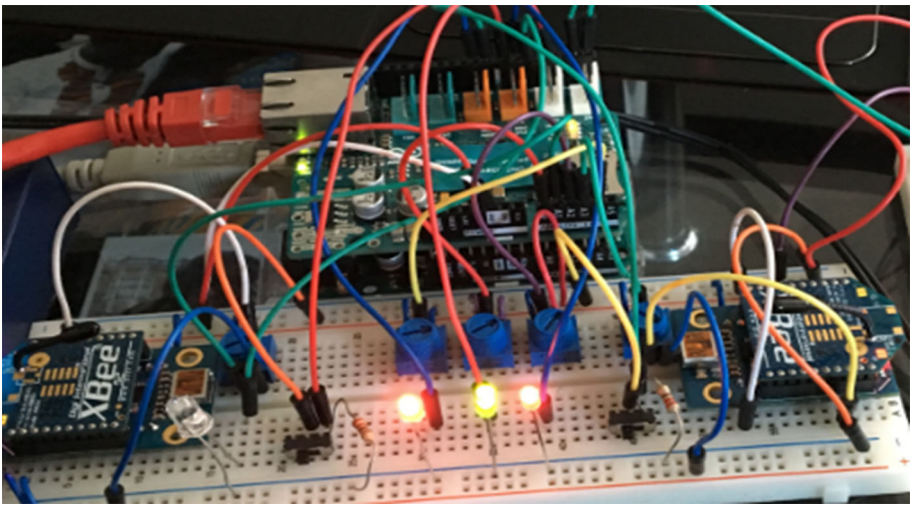


Fig. 6 Arduino board as a PLC

3.5 XBee Radio Modules

There are three XBee radio modules used for the simulation of the Smart Grid system (see Fig. 7). The XBees were used as simulation hardware for smart meters and a gateway in remote areas. The configuration of the radios was done with the XCTU software. Two of the radio modules were configured as XBee router radio modules and the other one as an XBee coordinator (gateway). The XBees were configured to have the same PAN IDs for exchange and routing of data to the Arduino (PLC).

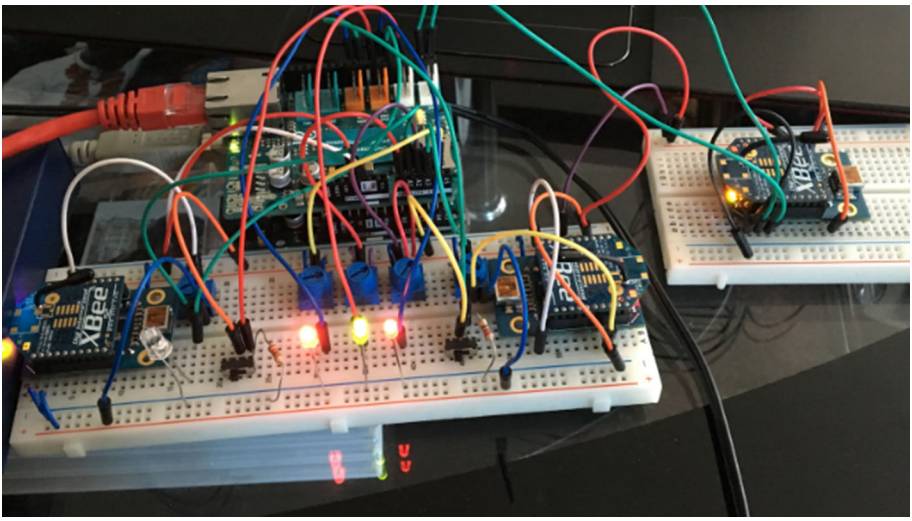


Fig. 7 XBee radio modules

The router radio modules were configured to send data to the coordinator radio by specifying a destination high address of 0013A200 and a destination low address as the address of the coordinator radio. For router radio modules to be able to read analog data from the potentiometers and send the data to the coordinator, the data input 0 (DIO 0) of the router radios were set as analog inputs using the XCTU software.

The XBee radio modules were configured as DigiMesh nodes to simplify the implementation of the Smart Grid. Digi XBee DigiMesh 2.4 [23] delivers end-point device connectivity with a globally deployable 2.4 GHz transceiver. The DigiMesh protocol is a peer-to-peer protocol that offers network stability through an effective network operation, self-healing and a low power consumption.

The coordinator radio module was connected to the Arduino via the serial communication port (TX and RX). The radio modules form a wireless PAN whereas the coordinator radio module and the Arduino form a wired area network.

3.6 Snort and Suricata (IDS and IPS)

Intrusion detection is a process of using a set of methods (detection system) to detect anomalous activity at the host and network level. IDS is software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems [10].

Snort is installed and configured to run as a NIDS by editing the configuration files (*snort.conf* and *local.rules*) and downloading community rules that Snort use to detect intrusion. The local rules configuration file can be customized for specific purposes. A typical Snort rule contains the rule header and the rule options. The rule header includes the rule's action (e.g., drop), protocol (e.g., tcp), source IP, source port, direction, destination IP, and destination port. The rule options consist of the alert message and information.

Snort can be run without root privileges by creating a user and group under which the daemon can run. The installation and configuration of Suricata is the same as Snort except that Suricata is configured to be used in a detection and prevention mode by dropping malicious packets.

For example, if an attacker attempts to send 100,000 tcp packets in 1 s to the Modbus default port (502), the Snort rule (sid: 10002) will be triggered, as shown in Fig. 8.

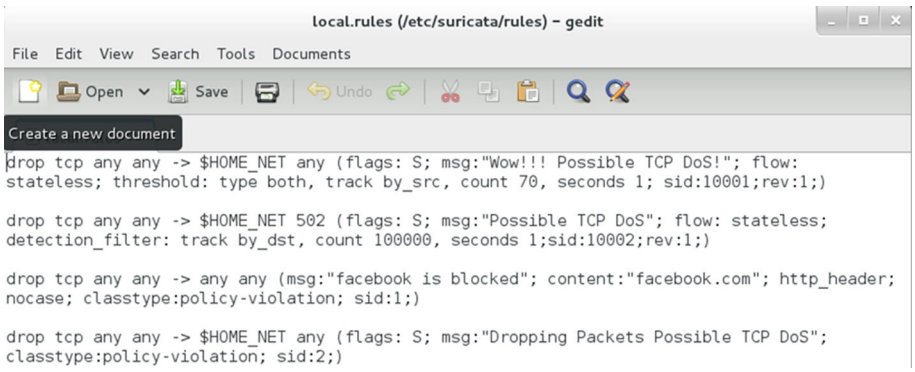


Fig. 8 Suricata (NIPS) local rules configuration file

3.7 BoNeSi (DDoS Simulator)

In this work, Bonesi is used to implement the DDoS attack. Bonesi is a very popular botnet simulator that can generate ICMP, UDP and TCP packets to flood a network in a test environment [18]. Bonesi is very powerful tools that can generate up to 150,000 packets per second (pps) and even more depending on the system hardware specification [18].

4 Results and Discussion

4.1 Flow of System Data

The first test was to examine data exchange between the grid and the subsystems. As shown in Fig. 9, 10 and 11, there was communication and flow of data from the Grid subsystems to the Smart Grid central server. This implied that the Modbus TCP protocol has been implemented successfully and functions as expected. A change in the data in the subsystems is transmitted to the Smart Grid central point and displayed in numerical mimics, measurement displays and graphical form. This set up the platform for carrying out the attack and implementing a defense mechanism.

Figure 9 shows the numeric values of the power generation and consumption of the various power units in kilowatts per hour (kWh) in white text boxes. The kWh value for the homes is an estimation of the average power consumption in the UK [24]. The annual estimation is used to provide a bigger data for the project. There are also gauges representing the various power units and the mains frequency. The mains frequency in the UK is 49.7 Hz and this was implemented to provide information when more power was needed to prevent an outage. There were red LEDs to indicate the flow of power to the homes.

Figure 10 shows the graphical view of the data coming from the homes, grid utility, generator and wind turbine. The trend was plotted as time against kilowatts per hour.

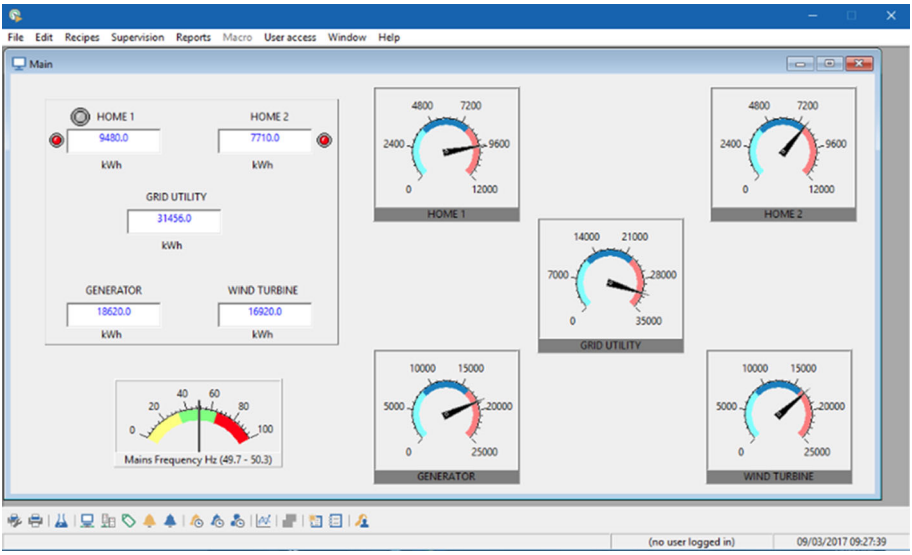


Fig. 9 SCADA server displaying receipt of data

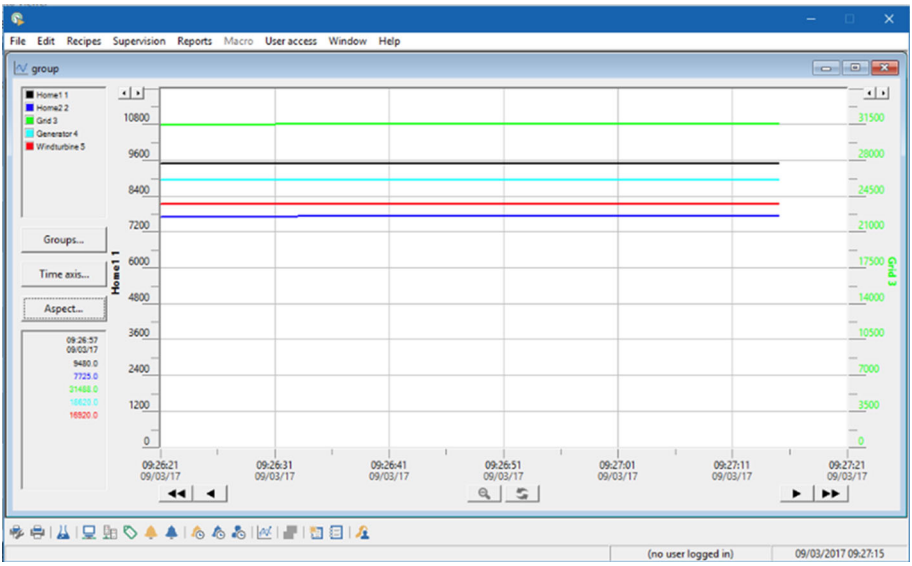


Fig. 10 SCADA server displaying data in a chart

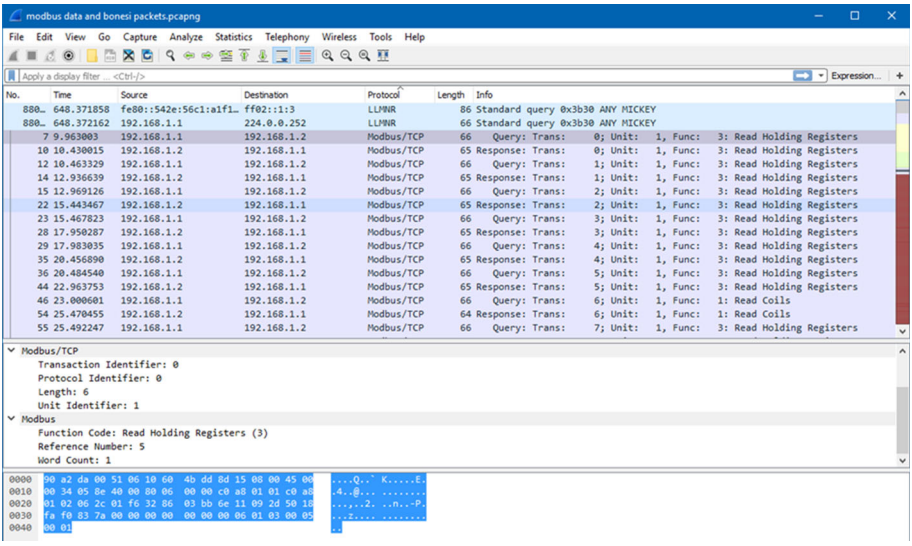


Fig. 11 Wireshark showing exchange of Modbus TCP request and response packets

Because this trend is real-time, it provides depth information to the control operator to monitor the system and analyze the data in real time.

When the SCADA server was up and running there were requests or queries for Modbus data sent from the server to the PLC holding the data from field devices. As shown in Fig. 11, the Modbus TCP query packets from the server (192.168.1.1) and response packets from the PLC (192.168.1.2) were captured in Wireshark. This proves the

successful implementation of the Modbus protocol as a communication mechanism in this study. The Modbus protocol running in default port 502.

4.2 Attack Implementation

DoS attack was chosen to reveal the vulnerabilities of the system and the extent of the failure of the system. As shown in Fig. 12, 13, 14, 15 and 16, the system could not function as expected due to the DoS attack. Attacking the SCADA server crashed the server (PC) leading to a Windows "Blue-Screen-of-Death." The second attack on a subsystem of the Smart Grid (PLC) caused an inconsistent response to requests from the SCADA server leading to device driver read errors, no data from subsystems and an unexpected fluctuation in data or intermittent communication between the Smart Grid server and the subsystems.

It can be observed in Fig. 12 that botnet attacks were created from different ports (sources) which produced 11,944 requests in less than 5 s. The large number of requests overwhelmed the PLC in just 5 s causing the PLC to stop responding to legitimate requests from the SCADA server. The PLC has allocated all of its resources to respond to the botnet attack requests.

Due to the attack, Fig. 13 shows the unanswered requests leading to fifty-nine read errors and a "KO" status of the Modbus device driver which stopped the driver from working (red highlighted text) although it was enabled.

The read errors had an effect of presenting no data to the SCADA database tags as shown in Fig. 14.

The unanswered requests resulting from the read errors caused a drop of communication between the server and the PLC for some time. Figure 15 shows the effects of the intermittent communication on the data from the various subsystems.

Figure 16 shows the exchange of packets (SYN and ACK) between an attack source (255.255.255.255) and the PLC (192.168.1.2). Every single packet of the thousands of packets on the wire was captured. Because the attack source did not acknowledge receipt

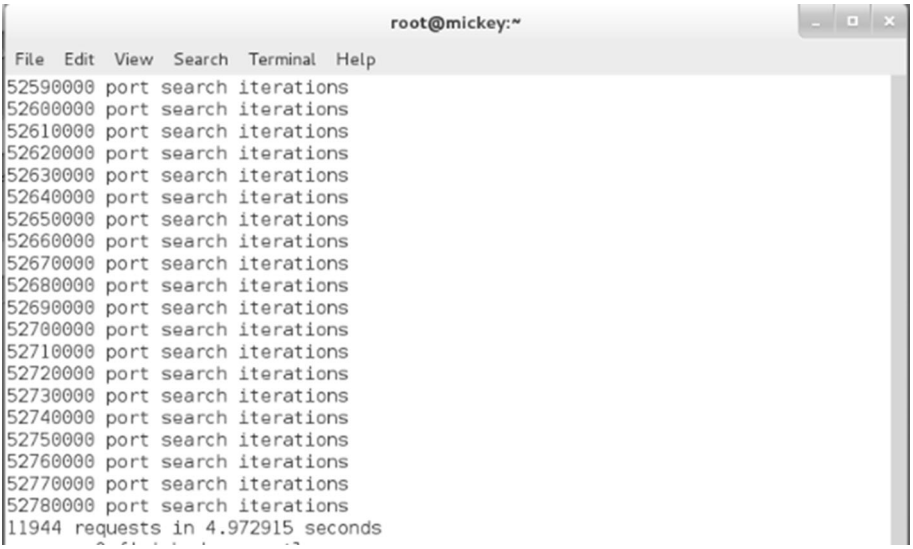


Fig. 12 Bonesi DDoS simulator flooding the network with SYN packets

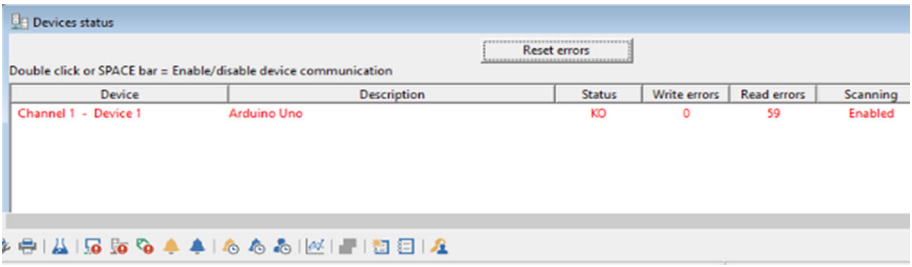


Fig. 13 SCADA server device driver read errors. (Color figure online)

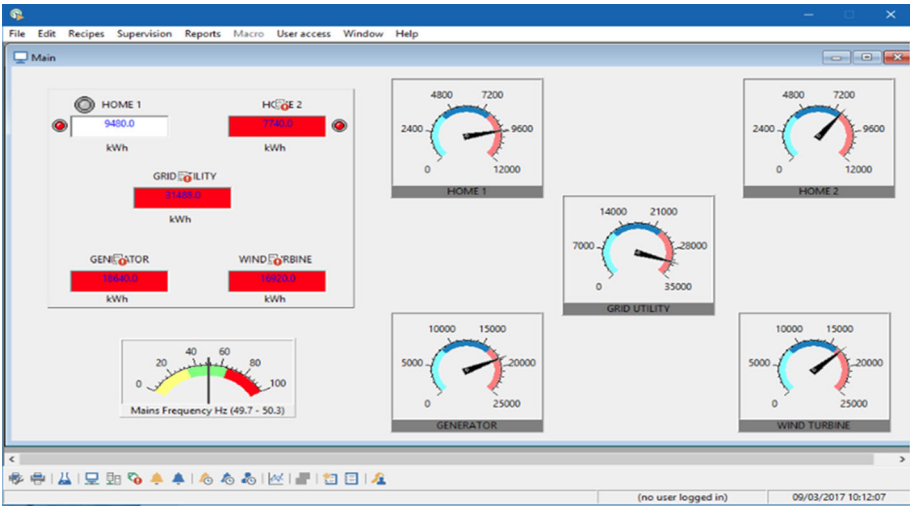


Fig. 14 Lost communication between Smart Grid server and subsystems

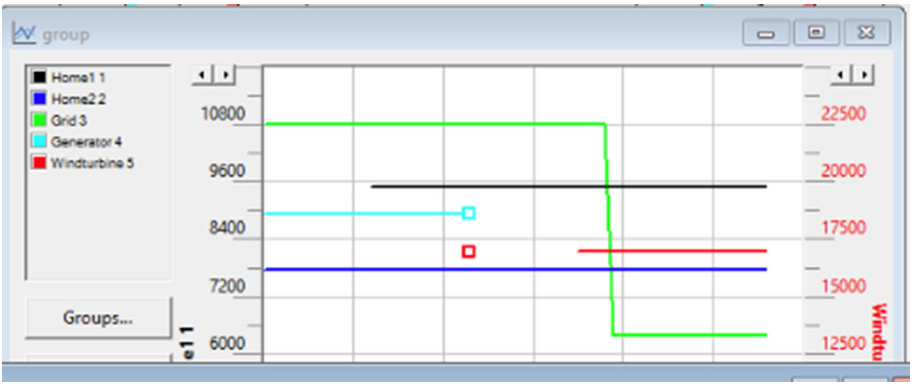


Fig. 15 Intermittent communication between systems

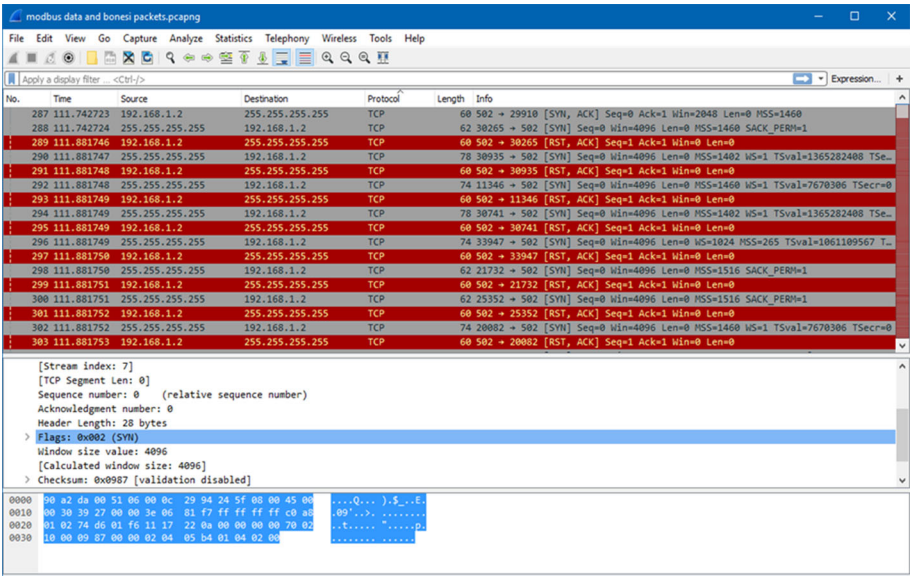


Fig. 16 Wireshark showing captured DoS SYN packets

of the data from the PLC it created a half-opened connection indicating to the PLC that the TCP connection should be kept alive for data retransmission. The open connection consumed the resources of the PLC causing it to reject the processing of requests from the SCADA server. It can be seen from the capture file (see Fig. 16) that there were no acknowledgements (ACK) packets from the attack device.

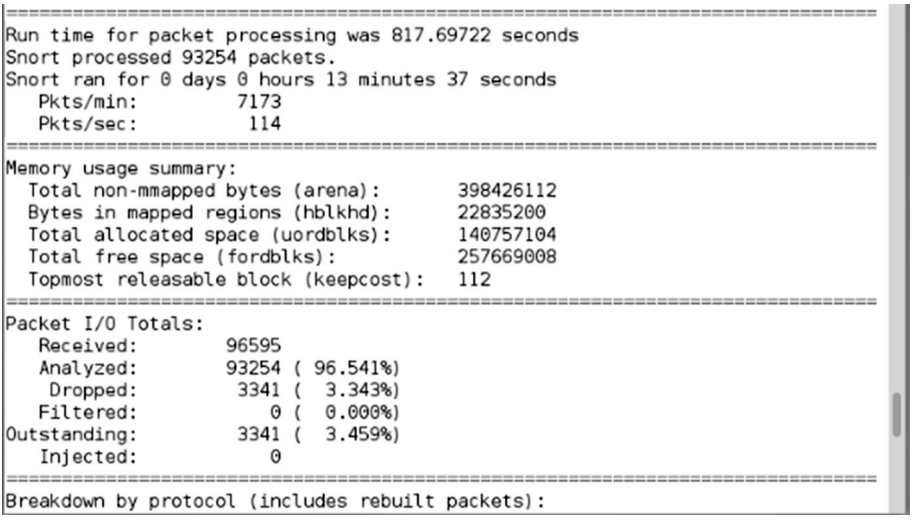


Fig. 17 Snort packet capture information

4.3 Cybersecurity Mitigation

After designing and building the Smart Grid testbed it was tested by attacking the various components of the testbed. An intrusion detection and prevention systems were deployed and it can be seen in Figs. 17, 18 and 19 that the system had been able to resist the cyber-attack by dropping the packets that were meant to flood the network. There was a steadily drop of packets at the initial stage because the NIPS could not drop all the attack packets in the fastest possible time or at the rate at which the requests were directed to the SCADA server and the PLC.

Figure 17 provides more information on the packet processing rate in seconds, run time of the NIDS, total packets received, analyzed and dropped, etc.

Figures 18 and 19 show the packets that were detected and dropped as a result of matching the DoS signature in the detection database of Snort and Suricata. The cyber-security mitigation system sniffed the packets, pre-processed the packets by checking them against plug-ins that determine the behavior of packets. The pre-processed packets were passed on to the detection engine with the plug-ins for a thorough check with the defined set of rules. The rule set can be continuously augmented with new rules as further malicious activities are detected.

As typical NIDS and NIPS implementation, when a match is found, an alert is raised, and the packet is dropped as observed in Figs. 18 and 19.

After the deployment of the cybersecurity mitigation with IDS and IPS, Figs. 20 and 21 show that the SCADA server and PLC have been able to operate continuously despite ongoing DoS attack.

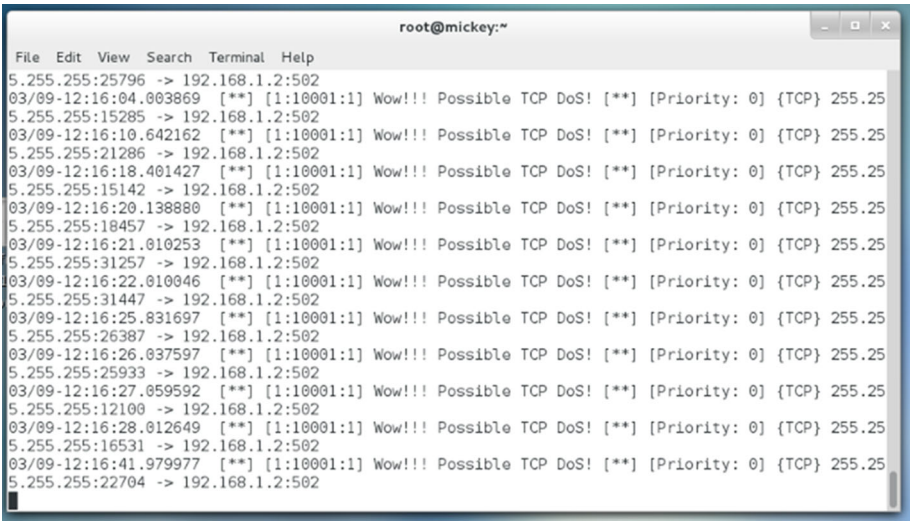


Fig. 18 Snort NIDS alerting intrusion detection

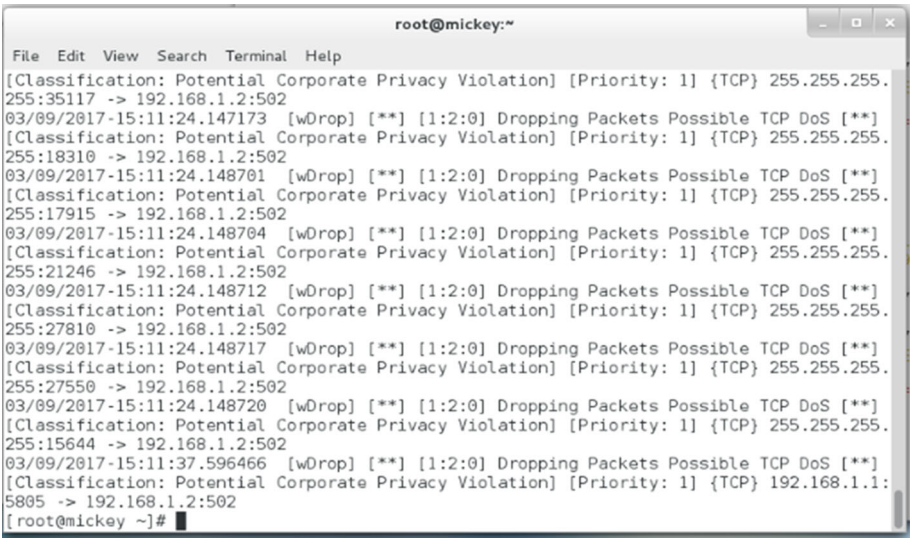


Fig. 19 Suricata NIPS dropping attack packets

Fig. 20 SCADA system running as expected despite ongoing attack

5 Summary

Cybersecurity is one of the emerging major threats in Smart Grid that needs to be considered as the attack surface increased. To prevent cyber-attacks, new techniques need to be evaluated in a real-world environment or in a testbed. However, the costs for setting-up Smart Grid testbed is extensive. The article has revealed the significance of protecting mission-critical systems from cyber-attacks. These SCADA systems are found in various sectors of the engineering world and provide economic and societal benefits to individuals and nations. The vulnerabilities of such systems need to be considered to provide defense mechanisms that would protect them.

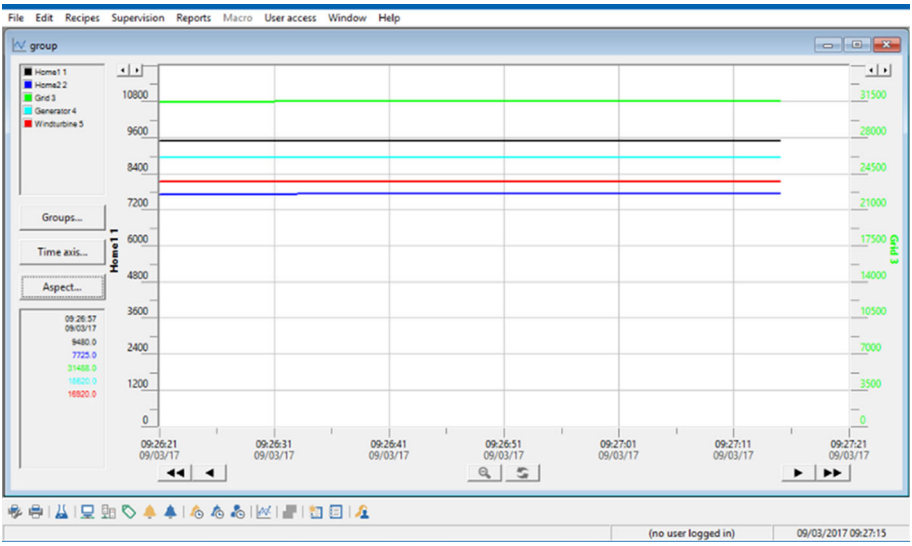


Fig. 21 SCADA system running as expected

In this study, we focused on the development of a Smart Grid testbed with a low-cost hardware and software for cybersecurity research and education. We proposed a low-cost Smart Grid testbed based on Arduino microcontroller. As a study case, DDoS cyber-attack was simulated and showed an expected effect on the system. The cyber-attack implementation was successful and the NIDS and NIPS were able to detect and prevent the attack after they were deployed. However, the IPS reacted partially to the attack at the initial stage due to the rate at which the attack happened took full control of the network, but the defense systems were eventually successfully blocked the attack after a few minutes once the attack packets were identified and processed more effectively. Furthermore, the testbed can be deployed with low amount of resources, providing educational benefits in terms of various scenarios that can be implemented and evaluated.

The work would be best continued by considering how the subsystems are implemented to provide a fully functioning Smart Grid infrastructure, other ways of attacking and protecting the system. This would require more advanced hardware and software for data generation and transmission, power distribution from the main grid and substations, different communication protocols, and more advanced cyber defense mechanisms such as machine learning based IDS and IPS.

References

1. Fang, X., Misra, S., Xue, G., & Yang, D. (2012). Smart Grid—The new and improved power grid: A survey. *IEEE Communications Surveys & Tutorials, 14,* 944–980.

2. Asri, S., & Pranggono, B. (2015). Impact of distributed denial-of-service attack on advanced metering infrastructure. *Wireless Personal Communications, 83,* 2211–2223.
3. Sun, C.-C., Liu, C.-C., & Xie, J. (2016). Cyber-physical system security of a power grid: State-of-the-art. *Electronics, 5,* 40.
4. Sun, C.-C., Hahn, A., & Liu, C.-C. (2018). Cyber security of a power grid: State-of-the-art. *International Journal of Electrical Power & Energy Systems, 99,* 45–56.
5. Andreadou, N., Olariaga Guardiola, M., Papaioannou, I., & Prettico, G. (2016). *Smart Grid Laboratories Inventory 2016*. Technical report, Joint Research Centre.
6. Bari, A., Jiang, J., Saad, W., & Jaekel, A. (2014). Challenges in the Smart Grid applications: An overview. *International Journal of Distributed Sensor Networks, 10,* 974682.
7. EPRI. (2007). *Advanced metering infrastructure (AMI)*. Palo Alto, CA: Electric Power Research Institute.
8. Y. Yang, K. McLaughlin, T. Littler, S. Sezer, E. G. Im, Z. Q. Yao, et al. (2012). Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in Smart Grid SCADA systems. In *International conference on sustainable power generation and supply (SUPERGEN 2012)* (pp. 1–8).
9. Yang, Y., McLaughlin, K., Sezer, S., Littler, T., Im, E. G., Pranggono, B., et al. (2014). Multiattribute SCADA-specific intrusion detection system for power networks. *IEEE Transactions on Power Delivery, 29,* 1092–1102.
10. Bace, R., & Mell, P. (2001). *Intrusion detection systems*. NIST Technical Report 800-31, National Institute of Standards and Technology (NIST).
11. Pranggono, B., McLaughlin, K., Yang, Y., & Sezer, S. (2014). Intrusion detection systems for critical infrastructure. In A.-S. K. Pathan (Ed.), *The state of the art in intrusion prevention and detection* (pp. 115–138). Boca Raton: CRC Press.
12. Roesch, M. (1999). Snort—Lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX conference on system administration* (pp. 229–238), Seattle, Washington.
13. Arabo, A., & Pranggono, B. (2013). Mobile malware and smart device security: Trends, challenges and solutions. In *2013 19th international conference on control systems and computer science (CSCS)* (pp. 526–531).
14. *Low orbit ion cannon*. https://sourceforge.net/projects/loic/. Accessed 10 April 2018.
15. *High orbit ion cannon*. https://sourceforge.net/projects/high-orbit-ion-cannon/. Accessed 10 April 2018.
16. Arduino. (2017). *Arduino UNO*. https://www.arduino.cc/. Accessed 21 April 2018.
17. Sielco Sistemi. (2017). *Winlog Lite (3.02.04 ed.)*. https://www.sielcosistemi.com/en/download/public/download.html. Accessed 21 April 2018.
18. Goldstein, M. (2008). *BoNeSi DDoS simulator*. https://github.com/Markus-Go/bonesi. Accessed 21 April 2018.
19. Suricata. (2017). *Open source IDS/IPS/NMS engine*. http://suricata-ids.org/. Accessed 21 April 2018.
20. Zigbee Alliance. (2017). *Zigbee*. http://www.zigbee.org/. Accessed 10 April, 2018.
21. Modbus-IDA. (2006). *Modbus messaging on TCP/IP implementation guide V1.0b*. http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf. Accessed 21 April 2018.
22. Zurawski, R. (2005). *The industrial communication technology handbook*. Boca Raton: CRC Press.
23. Digi.com. (2017). *Wireless mesh networking RF module*. https://www.digi.com/products/xbee-rf-solutions/embedded-rf-modules-modems/xbee-digimesh-2-4. Accessed 21 April 2018.
24. OFGEM. (2017). *Typical domestic consumption values*. https://www.ofgem.gov.uk/. Accessed 10 April 2018.

Michael Annor-Asante is a Control Systems Engineer (Diesel Engine and Turbocharger Test Systems Engineer) at Cummins Turbo Technology in Huddersfield, UK. Michael holds a degree in Computer Systems Engineering from Sheffield Hallam University and has worked on Cyber-security projects including a project on Intrusion Detection System for Power SCADA System (Smart Grid). Michael has professional experience in control systems, computer systems and engineering applications. He also has a Business Management and Accounting background which makes him unique as an Engineer. Before joining Cummins Turbo Technologies, Michael worked for National Grid UK (Servelec Controls) as a Software and Control Applications Engineer for nearly 2 years. When he isn't glued to a computer screen, he spends time doing his musical stuff and watching football or adventurous movies.

Bernardi Pranggono is a senior lecturer at the Department of Engineering and Mathematics, Sheffield Hallam University (SHU). Prior to joining SHU he was a lecturer at Glasgow Caledonian University. He held post-doctoral researcher at the Queen's University Belfast, where he worked on a range of EU and EPSRC projects in the Centre for Secure Information Technologies (CSIT). Previously, he held industrial positions at Accenture, Telstra, and PricewaterhouseCoopers. Dr. Pranggono received his B.Eng degree in Electronics and Telecommunication Engineering from Waseda University, Japan, M.DigComms degree in Digital Communications from Monash University, Australia and Ph.D. degree in Electronics and Electrical Engineering from the University of Leeds, UK. His current research interests include network security, optical networking, cloud computing, and green ICT. Dr. Pranggono has co-authored over two-dozen papers in leading international conferences and journals, and contributed to four book chapters. He has served as Vice-Chair and Technical Program Committee member in numerous international conferences, such as IEEE HPCC and GLOBECOM. He also serves as referee of some renowned journals and conferences, such as IEEE Transaction on Industrial Informatics, IEEE Transaction on Power Delivery, IEEE Communication Magazine, IEEE Computer, IEEE GLOBECOM, IEEE ICC, Elsevier Optical Switching and Networking, etc.