SCADA-SST: A SCADA Security Testbed

Asem Ghaleb
Information and Computer Science Department
KFUPM, Dhahran, 31261, KSA
Email: g201305010@kfupm.edu.sa

Sami Zhioua
Information and Computer Science Department
KFUPM, Dhahran, 31261, KSA
Email: zhioua@kfupm.edu.sa

Ahmad Almulhem
Computer Engineering Department
KFUPM, Dhahran, 31261, KSA
Email: ahmadsm@kfupm.edu.sa

***Abstract*—The number of reported cybersecurity incidents on SCADA (Supervisory Control and Data Acquisition) systems increased significantly in the past few years. One contributing factor is the fact that security testing of live SCADA systems is not practical as such systems are expected to be operational 24/7. Also and most importantly, conducting live security testing on these types of systems is generally costly. A practical and cost-effective solution is to carry out security testing on a simulated version of the physical setting. The main contribution of this paper is to present a SCADA simulation environment (SCADA-SST) suitable for security testing. The simulation environment is generic, easy to setup (comes with a detailed manual), and supports hybrid architectures (involving simulated as well as physical components). We show how SCADA-SST can be used to simulate two realistic settings, namely, Water distribution and Electrical power grid. Finally, for the sake of security testing example, we show how SCADA-SST can be used to assess the resilience of common SCADA nodes to DOS attacks.**

***Keywords*-SCADA; Simulation; Security testing; Industrial Control Systems; Network attacks;**

I. Introduction

Historically, the security of SCADA systems relied on two forms of protection. The first is employing an air gap, that is, the SCADA network would be physically isolated from any other network, thus making it harder for an attacker to gain access. The second is security through obscurity, that is, vendors and operators relied on the fact that very little, if any, information was publicly available about their environments. Hence, SCADA protection focused on restricting access to unmanned field networks and preventing configuration mistakes.

In recent years, most SCADA systems became connected (directly or indirectly) to the Internet. This can be explained by the need for sharing real-time information with the business operations which is now a necessity in order to improve efficiency, minimize costs, and maximize profits. This, however, exposes SCADA systems to various types of exploitation. Analyzing and improving the security of SCADA systems require security evaluation and testing. These tasks cannot be performed while the systems are operational: they might lead to system failure and downtime while SCADA systems are expected to be up and working 24/7. Setting up a second physical system as a clone for the operational SCADA system is not a common practice mainly because of its cost.

A common alternative for SCADA security testing is to simulate the physical setting in a virtual environment and to carry out all the evaluation and testing tasks on the simulated version. In addition of being very cost-effective, this alternative allows to switch quickly from one topology/architecture to another making it possible to experiment with several security/protection scenarios.

In this paper, we present SCADA-SST[1], a SCADA security simulation environment. SCADA-SST is characterized by three attractive features:

1) **Generic**: It is designed to be generic enough to be used in various scenarios and settings specific to different fields.
2) **Lightweight**: The overhead of the simulated nodes is minimal making it possible to represent very large SCADA systems.
3) **Supports hybrid scenarios**: It allows hybrid configurations involving simulated as well as physical components.

SCADA-SST has been designed and implemented in order to facilitate SCADA security evaulation and testing. The environment is available for public use [13] and comes with an easy to understand manual. As a demonstration, we show how SCADA-SST can be used to simulate two concrete use cases: water tanks control and smart-grid.

II. Related Work

Only few SCADA simulation environments have been proposed in the literature [4], [9], [1], [5], [6]. In this section, we briefly describe existing SCADA simulation environments and point out the limitations of each. This should provide, we hope, enough justification to propose a new SCADA simulation environment for security testing.

McDonald et al. [9] described a virtual control system environment developed at Sandia National Laboratories for investigating SCADA vulnerabilities in the field of energy systems. The virtual control system is composed of simulated RTUs that interact with simulated power systems

[1]SST stands for SCADA Security Testbed.

978-1-908320-63/6/$31.00 ©2016 IEEE

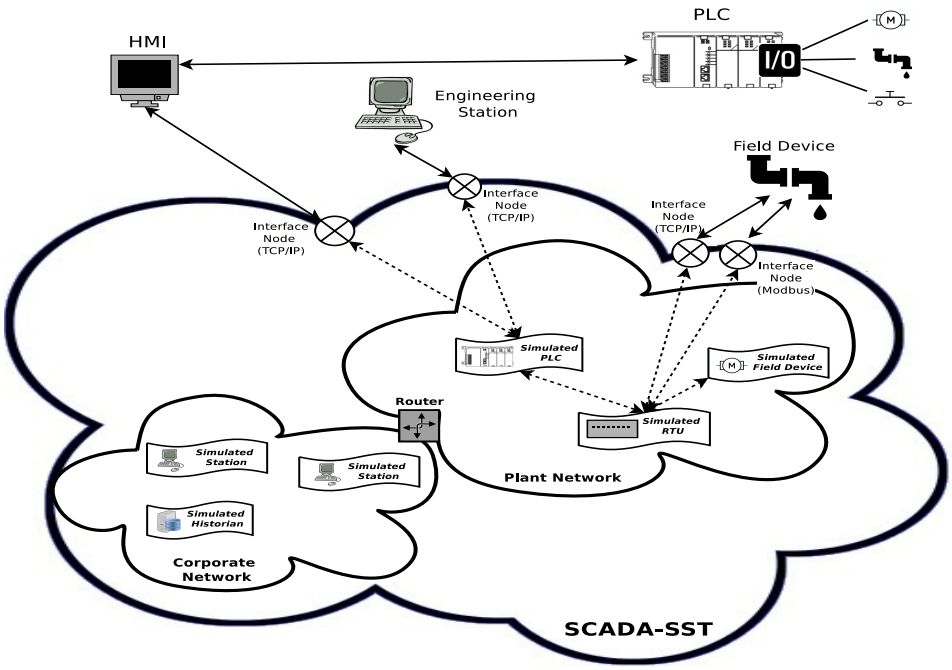

Figure 1. SCADA-SST Components

(PowerWorld SimAuto server [15]). Although the environment is suitable to simulate power systems, it is not generic enough to be used in other fields. Besides, some of the simulation environment are proprietary (e.g. PowerWorld SimAuto server), hence not available for public use.

Chabukswar et al. [4] used the C2WT (Command and Control Wind-Tunnel) [7] framework to simulate DDOS-like attacks on a plant and its control system as well as to analyze the effects on different routers. The C2WT framework is based on HLA (High Level Architecture) and is typically used to facilitate the development of large-scale simulations. Chabukswar et al. used the NetworkSim, which is based on OMNeT++ to simulate the communication protocols and the Simulink to model the domain specific processes. They developed Simulink functions to synchronize the model with the Run-Time Infrastructure allowing Simulink to progress only when the RTI allows it. They used timed-stepped synchronization, while keeping an appropriate small time-step size in order to minimize event timing errors introduced by exchanging events between Simulink and HLA. However, similarly to the work of McDonald et al., this C2WT-based simulation environment is designed mainly for power plant simulation and specific type of scenarios (i.e. DDOS attacks).

SCADASim [1] is a simulation tool developed for security testing. The main purpose of the tool is to test specific attacks and to examine their effects on real devices and applications. Supported attacks include denial of service, man in the middle, eavesdropping, and spoofing. They used the OMNeT++ to simulate the network and they exploited the socket based integration of OMNeT++ to allow the integration of the external devices using deployed gates. The main limitation of SCADASim is that it is limited to simulated attacks, that is, it does not provide a framework to launch attacks from outside the simulation framework. The hardware components were simulated using MATLAB/Simulink but no real physical hardware was mentioned except sensors and actuators.

III. SCADA-SST

SCADA-SST is a simulation testbed for SCADA systems that allows users to easily build network topologies using drag and drop and with minimal amount of coding. It is implemented using OMNeT++ network simulator [16] in combination with INET framework [17] that contains all libraries needed to build communication network models and implements the most common Internet protocols, such as TCP, IP, UDP, MAC protocols, etc.

Simulated SCADA systems are built by creating SCADA-SST components within a topology. The behavior of SCADA-SST components is written in C++ which allows dynamic binding of the C++ programs describing the behavior of the components. The SCADA-SST is then tuned by setting some topology configuration parameters. The components communicate with each other via simulated network packets and make decisions based on the packets they receive.

978-1-908320-63/6/$31.00 ©2016 IEEE

A. Architecture and components

The general architecture as well as the different components of SCADA-SST are illustrated in Fig. 1.

There are three types of nodes in SCADA-SST, namely, simulated nodes, external (real) nodes, and interface nodes. Simulated nodes are SCADA related devices (e.g. PLC, RTU, HMI, etc.) that are simulated using OMNet++. External nodes are physical SCADA devices that are located outside the OMNet++ simulated network. Interface nodes allow communication between the simulated nodes and the external nodes. All these components can be placed in any desired network topology.

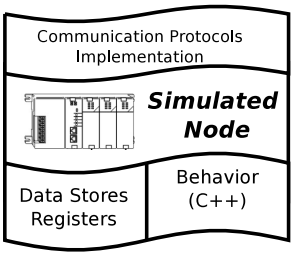

Figure 2. Simulation Node Components

1) Simulated Node: A simulated node is an instance of an OMNet++ node with SCADA operation features. It consists of three software constructs, namely, node behavior, data stores/registers, and communication protocols implementation (Fig. 2). The node behavior is implemented through a C++ program that specifies how the node should behave, process input and produce output from/to other SCADA-SST nodes. The C++ behavior program simulates the logic and processing normally carried out by a physical device.

The second software construct, namely, the data stores and registers are used to store relevant constrol system related data values and parameters[2]. These values will be available to the current simulated node as well as to the other attached simulated nodes for read and write commands (register_read/register_write).

The third software construct is the communication protocol implementation which allows the simulated node to communicate with other SCADA-SST components. Each simulated node contains an instance of the communication protocol implementation which specifies how to process incoming packets and how to construct outgoing packets.

2) Interface Node: One of the design requirements of the SCADA-SST simulation environment is to allow external real devices to be connected to the simulated network as any other simulated node. The integration of external real devices and applications in OMNet++ can be achieved in three possible ways, namely, source code integration, shared library integration, or through the use of socket connections.In SCADA-SST, the integration is implemented using socket connections in a special type of node called interface node (Fig. 3). An interface node sits between every pair of simulated and external nodes that need to communicate.

[2]Data stores and registers are commonly used in industrial control system devices to store relevant parameter values locally so that to reduce the number of times these values are fetched from other devices.

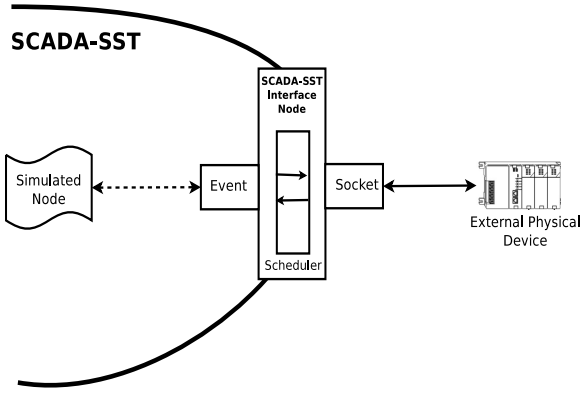

Figure 3. SCADA-SST Interface Node

3) External Node: An external node is a real device connected to SCADA-SST through a real communication media. This type of nodes allows SCADA-SST to simulate hybrid configurations involving both simulated and real devices. Communication with an external device is possible provided that SCADA-SST has an implementation of the protocol used by the external device. Communication is then achieved through an interface node. An external node can be an external application representing an HMI.

IV. Use Cases

In this section, we show how SCADA-SST can be used to simulate two realistic SCADA systems. The two use cases cover two different fields, namely, water distribution and electric smart grid and are both hybrid scenarios involving both simulated and physical nodes.

A. Water Distribution

Fig. 4 shows the first use case which is a SCADA system used to control water tanks in a water distribution station. As shown in the figure, there are two water tanks. The first tank (Tank1) is filled from the water source and then used to distribute water. The second tank (Tank2) is used as an extra tank to reduce the pressure on Tank1. In case the water level in Tank1 reaches a pre-specified level, a transfer pump

978-1-908320-63/6/$31.00 ©2016 IEEE

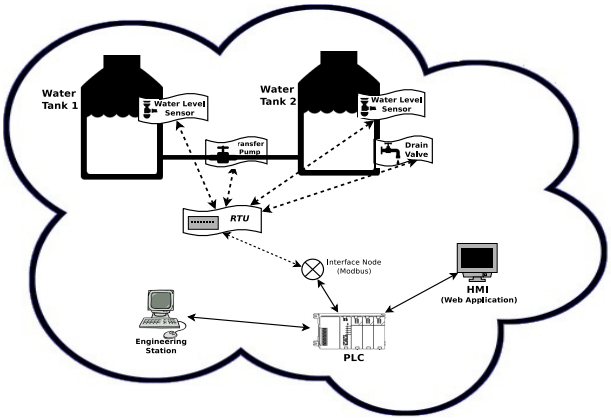

Figure 4. Water Distribution Use Case

between Tank1 and Tank2 would be opened to pass the water from Tank1 to Tank2. In case the water level in both tanks exceeds the permitted water level, a drain valve on Tank2 would be opened to dispense water. The configuration details of each component is described as follows.

- **PLC**: It is a real Siemens SIMATIC S7-400 PLC that has been integrated with the simulated components using modbus/tcp protocol. It is configured to work as a master device to (1) collect data from the simulated RTU and then sends it to the HMI and to (2) forward operator's commands back to the RTU.
- **RTU**: It is a simulated node using SCADA-SST RTU node template. It gathers information from the tanks' sensors and works as a slave device listening on port 502 for coming requests from the PLC.
- **Engineering Station**: It is Windows 7 host machine equipped with Siemens Simatic PCS7 V8.0 software which is a programming and configuration environment for Siemens PLC. In this scenario, it is used to configure the physical PLC and connect it with the simulated components.
- **Water-Level Sensors**: These are SCADA-SST simulated nodes that represent water-level sensors in both tanks. These simulated node will generate values according to a pre-established model. The two sensors are connected to the RTU through modbus protocol.
- **HMI**: It is an external node implemented as a web-based application. It receives updates about the status of the water tanks from the PLC and presents it in graphical way through the web interface. It allows operators to interact with the controlled field devices by starting/stopping transfer pump or opening/closing the drain valve, etc.

B. Smart Grid

The second use case is a smart grid system that is used to control electricity consumption in "smart" homes (Fig. 5). For simplicity, the scenario involves two houses. Each house is equipped with a smart electric meter that measures the house's power consumption. The power consumption data is collected by an RTU and then sent to the electric company through Internet. Based on the collected data that is processed by the MTU, the system operators (viewing the system status through HMI) would send commands to adjust the electric power supply in order to optimize the distribution.

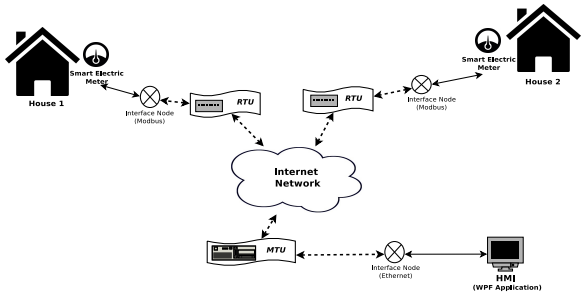

Figure 5. Electricity Smart Grid Use Case

The configuration details of each component is described as follows.

- **MTU**: It is a SCADA-SST simulated node created using the MTU template. It is configured to work as a master device to collect data from the RTUs through Internet. The processed data is then sent to the HMI.
- **RTU**: It is a SCADA-SST simulated node (RTU node template). It gathers information from the house's electric meter and works as a slave device listening on port 502 for coming requests from the MTU. It is connected to the smart electric meter through a Modbus interface node.
- **Electric Meter**: It is an external node representing a smart meter. It is implemented as a program that simulates power consumption values following a realistic pattern.
- **HMI**: It is a Microsoft WPF application representing an HMI. It receives updates about houses power consumption through the MTU and presents them to the operators in a graphical way. Since the HMI is an external node, it is connected to the MTU through an Ethernet interface node.

978-1-908320-63/6/$31.00 ©2016 IEEE

V. SCADA-SST Security Testing Features

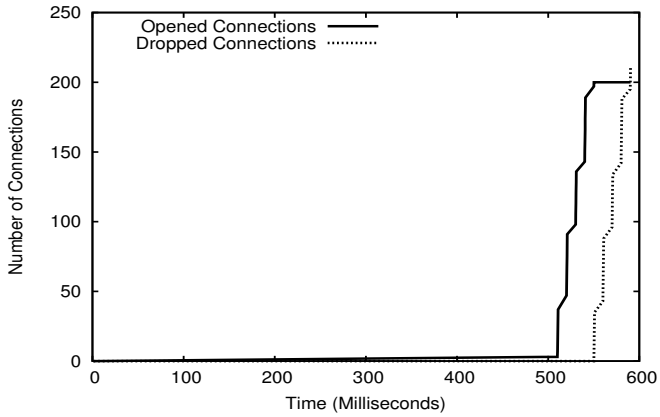

Figure 6. Number of opened and dropped connections during a denial of service attack launched from a single host

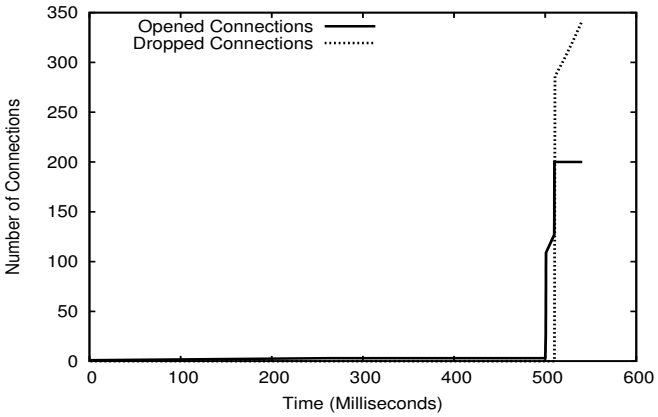

Figure 7. Number of opened and dropped connections during a denial of service attack launched from 15 malicious hosts

SCADA-SST is designed mainly for security testing and analysis of SCADA and industrial control systems without interfering with daily operations. The goal is to identify vulnerabilities and cybersecurity deficiencies in SCADA systems at an early stage. One approach to achieve this goal is to implement exploits and attack the simulated SCADA system. This allows to understand the implications of SCADA vulnerabilities, develop taxonomies of cybersecurity dificiencies, and ultimately develop cybersecurity mitigations which can be implemented and validated before deployment of the SCADA solution.

SCADA-SST contains several features that facilitate common security testing operations. These include malicious nodes templates, network attack scenarios, communication traffic capture and analysis capabilities, etc.

- **Traffic Capture**: Capturing network traffic is essential to detect network intrusions and attack patterns. SCADA-SST defines a particular type of nodes that can be configured to capture part or all packets in the simulated network. The template, called *TCPDump*, is based on tcpdump packet sniffer and analyzer [18].
- **Network Attacks**: SCADA-SST defines several node templates that implement typical network attacks. This include:
  - DOS zombie node: implements a denial of service attack on selected simulated nodes.
  - TCP spoofing host: implements a TCP spoofing attack capabilities which consists in replacing the source address in certain packets to make them appear as coming from other nodes.
  - Worm host: implements worm based attack on all the simulated network.

A very common network attack on SCADA systems is denial of service which targets the availability of certain nodes. Using the water distribution use case (Section IV-A), we simulated a DOS attack to illustrate the DOS zombie and traffic capture features of SCADA-SST. The attack scenario consists in creating one or several DOS zombie nodes and then use them to flood the simulated RTU node (Fig. 4). The RTU is in charge of gathering information from the the tanks' sensors and executing commands received from the PLC. Any delay in sending and/or processing information from/to the sensors and the PLC may lead to serious disturbance in all the SCADA system.

Two attacks have been simulated. The first one is a simple DOS attack involving one single DOS zombie node. The second one is a DDOS attack where 15 DOS zombie nodes are used to target the RTU. Both attacks use a typical TCP SYN flood.

To assess the consequence of each attack and its impact on the behavior of the RTU, we tracked the number of opened and dropped connections before, during, and after the DOS attacks. A connection is considered dropped if the RTU does not send an ACK back.

The RTU is configured to maintain a maximum of 200 simulatenous connections. When the number of opened connections reaches this limit, the RTU starts dropping incoming connections. The dropped connections might be legitimate connections with the tanks' sensors and/or the PLC.

Fig. 6 shows the number of opened and dropped connections during the DOS attack (one single attacking node).

978-1-908320-63/6/$31.00 ©2016 IEEE

The attack is scheduled to start exactly 500 seconds after the beginning of the simulation session. Right after the starting of the attack, the number of opened connections rises gradually until it reaches the threshold of 200. From that point on, the number of dropped connections starts to rise according to the same pattern as the opened connections. Since the flood is launched from a single DOS zombie node, there is a 50 seconds delay between the start of the attack (500 seconds) and the start of the connections droppings (550 seconds).

In the DDOS attack scenario, 15 DOS zombie nodes are used to flood the same RTU simulated node. As illustrated in Fig. 7, the time delay between the starting of the attack (500 seconds) and the beginning of the connections droppings is much smaller. That is, the RTU starts to drop connections soon after the launching of the attack because the 200 threshold is reached very quickly.

Such attack simulation shows how the SCADA-SST can be used to carefully analyze the resilience of proprietary SCADA nodes to common network attacks. The attack scenarios can be fine-tuned to test, ahead of time (before actual deployment), very interesting and very specific situations that may arise in SCADA deployments.

VI. Conclusion

In this paper we propose SCADA-SST, a SCADA security simulation environment that is generic (can be used in various fields), scalable (can simulate a large number of nodes), and supports hybrid scenarios (involving both simulated and physical nodes). SCADA-SST comes with several features that facilitate common security analysis operations such as malicious nodes templates, network attack scenarios, network traffic capture and analysis. Using a realistic water distribution use case, we showed how SCADA-SST can be used to assess the resilience of RTU implementation to a DOS and DDOS attacks.

Acknowledgment

This research was supported by The National Science, Technology and Innovation Plan (NSTIP) grant, NSTIP 13-INF281-04 at King Fahd University of Petroleum and Minerals.

References

[1] C. Queiroz, A. Mahmood, Z. Tari: 'SCADASim a framework for building SCADA simulations', IEEE Transactions on Smart Grid, 2011, 2, (4), pp. 589 597

[2] Maglaras, Leandros A., Jianmin Jiang, and Tiago Cruz: 'Integrated OCSVM mechanism for intrusion detection in SCADA systems', IET Electronics Letters, 2014, 50, (25), pp. 1935-1936

[3] Brown, Tom: 'Security in SCADA systems: How to handle the growing menace to process automation', IET Computing and Control Engineering Journal, 2005, 16, (3), pp. 42-47

[4] R. Chabukswar, B. Sinopoli, G. Karsai, A. Giani, H. Neema, A. Davis 'Simulation of network attacks on SCADA systems', First Workshop on Secure Control Systems, 2010.

[5] W. Chunlei, F. Lan, D. Yiqi: 'A simulation environment for SCADA secu- rity analysis and assessment', 2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), (1), IEEE, 2010, pp 342347

[6] A. Almalawi, Z. Tari, I. Khalil, A. Fahad: 'SCADAVT-a framework for SCADA security testbed based on virtualization technology', in 2013 IEEE 38th Conference on Local Computer Networks (LCN), IEEE, 2013, pp. 639646

[7] K. E. Roth, S. K. Barrett: 'Command and control wind tunnel integration and overview', in Proceedings of the 2009 SISO European Simulation In- teroperability Workshop, Society for Modeling and Simulation International, 2009, pp. 4551

[8] J. Ahrenholz: 'Comparison of core network emulation platforms', in: 2010- MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE, 2010

[9] McDonald M. J., Conrad, Service T., Cassidy R.: 'Cyber Effects Analysis Using VCSE-Promoting Control System Reliability', Andia National Laboratories Report (SAND2008-5954), 2008

[10] G. Deconinck, H. Beitollahi, G. Dondossola, F. Garrone, T. Rigole: 'Testbed deployment of representative control algorithms', Deliverable D9, Project CRUTIAL EC IST-FP6-STREP, 27513, 2008

[11] L. A. Rossman: 'Epanet 2: users manual', US Environmental Protection Agency. Office of Research and Development. National Risk Management Research Laboratory

[12] 'Dell security annual threat report', https://software.dell.com, Accessed May-2015

[13] 'SCADA-SST Sourceforge Project', https://sourceforge.net/projects/scada-sst, Accessed November-2016

[14] 'National Laboratory, National scada test bed program', https://www.inl.gov/scada/, Accessed March-2015

[15] 'Powerworld automation server (simauto)', http://www.powerworld.com/products/simulator/add-ons-2/simauto, Accessed August-2015

[16] 'OMNET++ Discrete Event Simulator', https://omnetpp.org, Accessed January-2016

[17] 'INET framework 2.1 for OMNeT++ Manual, version 4.6', https://inet.omnetpp.org/, Accessed December-2015

[18] 'Tcpdump and libpcap', http://www.tcpdump.org, Accessed June-2015

[19] 'The Center for SCADA Security Sandia National Labs, National scada testbed', http://energy.sandia.gov, Accessed March-2015

[20] 'Open source c library of modbus protocol for linux, mac os x, freebsd, qnx and win32, v3.0.6', http://libmodbus.org, Accessed August-2015