A Testbed for Secure and Robust SCADA Systems

Annarita Giani, Gabor Karsai, Tanya Roosta, Aakash Shah, Bruno Sinopoli, Jon Wiley[*][†][‡][§]

Abstract

The Supervisory Control and Data Acquisition System (SCADA) monitor and control real-time systems. SCADA systems are the backbone of the critical infrastructure, and any compromise in their security can have grave consequences. Therefore, there is a need to have a SCADA testbed for checking vulnerabilities and validating security solutions. In this paper we develop such a SCADA testbed.

1 Introduction

SCADA refers to a large-scale, distributed measurement (and control) system. The supervisory control system is placed on top of a real time control system to control an external process. SCADA systems are used to monitor or to control chemical or transport processes, in municipal water supply systems, to control electric power generation, transmission and distribution, gas and oil pipelines, and other distributed processes.

SCADA systems are comprised of three components:

1) *Remote Terminal Units (RTU)*: connects to the physical equipment and collects the bulk of the data. The RTUs must provide data reliability and data security.

2) *Master station and Human Machine Interface (HMI)*: consists of the servers and software that connect to the field equipment. HMI is responsible for compiling and formatting the collected data so that the human operator can make appropriate supervisory control decisions.

3) *Communication infrastructure*: used to connect various components of the SCADA system together. This infrastructure consists of, for example, multiplexed fiber-optic, satellite network, and Internet.

More details of these components will be given in Section 2. Given the critical nature of the SCADA systems, ensuring their security is of great importance. Attacks on the SCADA system can have serious consequences, such as endangerment of public health and safety, environmental damage, and significant financial impacts. There is a growing interest that the current SCADA systems are vulnerable to many cyber attacks [14]. Protection of SCADA systems has traditionally been based on the security by the obscurity concept. Proprietary protocols prevent an attacker from breaking into the system due to insufficient knowledge. Today such protection relies mainly on standards, recommendations, policies, and suggestions for possible countermeasures [1]. In order to better understand how to protect SCADA systems, it is imperative to perform vulnerability assessment on these systems and develop appropriate security mechanisms to protect the SCADA systems against attacks. To do so, developing a SCADA system testbed is essential. Recently, a SCADA testbed for the power system has been developed in [18]. Sandia National Laboratories SCADA testbed [4] is an example of a government sponsored testbed. The European community has also started working on creating a SCADA security testbed [5].

In this paper, we describe our SCADA security testbed. The rest of the paper is organized as follows: Section 2 discusses the reference architecture for the SCADA testbed. Section 3 explains the testbed implementation of our system in detail. Section 4 discusses the attack scenarios we plan to perform on the SCADA testbed. Sections 5 and 6 describe the status of the SCADA testbed and the next steps in the process. Section 7 concludes the paper.

2 Reference Architecture

In this section we detail the functional layers of our SCADA testbed architecture and discuss the interactions between them. Figure 1 shows the reference architecture for this testbed.

The corporate network represents the business end of an utility. This network is typical of an enterprize with a LAN/WAN connected to the Internet. However, in the case of utilities and industrial plants, the corporate network is often connected to the SCADA network in order to simplify business processes by allowing network

[*] The author list is alphabetical.

[†] A. Giani and T. Roosta are with the Department of Electrical Engineering and Computer Science, UC Berkeley {agiani,roosta}@eecs.berkeley.edu

[‡] A. Shah and B. Sinopoli are with the Department of Electrical and Computer Engineering, Carnegie Mellon University aakashs@andrew.cmu.edu, brunos@ece.cmu.edu

[§] J. Wiley and G. Karsai are with the Department of Electrical and Computer Engineering, Vanderbilt University {wileyjm,gabor.karsai}@isis.vanderbilt.edu

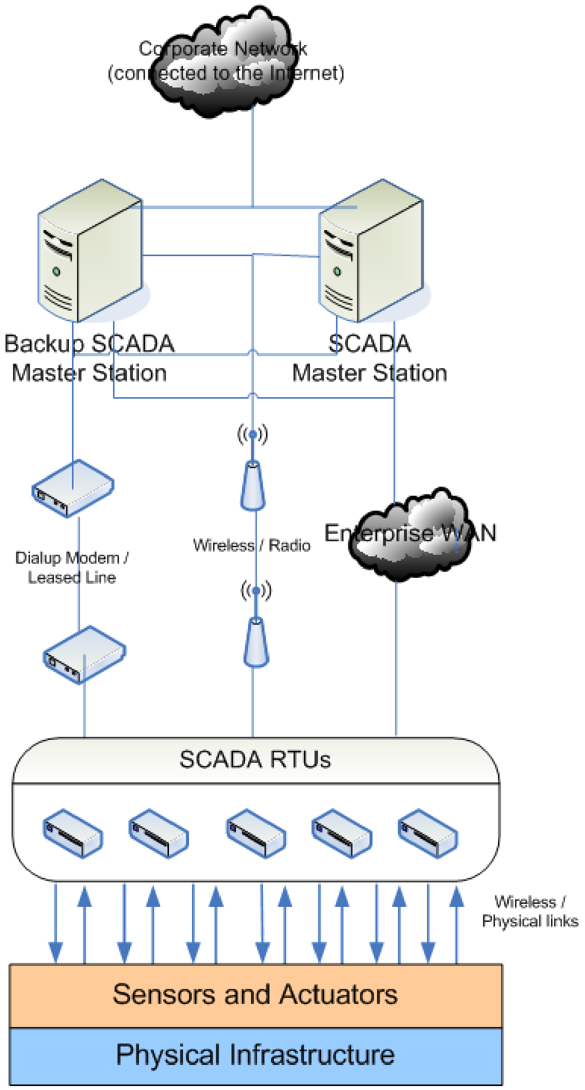

Figure 1: Reference Architecture

access to critical data on SCADA servers. This is one of the biggest information assurance concerns related to SCADA systems as an attacker can now connect to the SCADA network via the Internet by compromising nodes on the corporate network.

The SCADA master station consists of the SCADA master servers and the HMI. The master station is located in a central control center from where operators can monitor the performance of the entire system. SCADA master servers run the server side applications that communicate with the RTUs. The SCADA master servers poll the RTUs for data and send control messages to supervise and control the utility's physical infrastructure. Backup servers are used to increase fault-tolerance of the system. In order to add resilience, a backup master station may also reside in a physically separate location with independent communications channels to the RTUs. Various backup configurations may be used including hot, warm and cold backups.

Figure 1 also shows the various communication media commonly seen in a SCADA network. Dial-up modem, private leased line, wireless/radio and LAN/WAN links are widely used. From a SCADA system perspective, the primary difference between these links is generally the speed of communication and the noise on the channel. The communication protocols used over these channels vary based on the RTUs. There exist hundreds of different SCADA protocols, many of which are proprietary. However, Modbus (RTU, ASCII or TCP) [16] and DNP3 [7] are by far the most prevalent. Almost all SCADA protocols lack any authentication or confidentiality mehcanisms, making these communications channels vulnerable to attacks.

A utility may have anywhere from hundreds to thousands of RTUs controlling its infrastructure. RTUs are generally physically distant from the SCADA control center and can be miles away. In many cases, the RTUs are not physically secured. Most RTUs (especially legacy units) do not have proper information security mechanisms. Passwords are often sent in the clear and there is no way to authenticate the SCADA master server. RTUs have analog and digital I/O that interface with sensors and actuators connected to the infrastructure. This interface can be wired or wireless. Wireless HART [11] is an example of a wireless communications protocol used by RTUs to communicate with the sensors and actuators. The RTUs may be configured in a variety of different network topologies. The link between the SCADA master server and RTUs may be point-to-point or point-to-multipoint. The RTUs may themselves be configured in a cascading topology as well.

The physical infrastructure can represent the power grid, natural gas distribution/transmission system, water distribution system etc. It is the infrastructure being controlled and monitored by the SCADA system. SCADA systems may regulate the pressure of the gas/water pipeline or the voltage in the electric power grid. Sensors and actuators connected to the RTUs are placed along various points of the infrastructure in order to effectively perform this task. In many cases, the physical infrastructure has significant redundancy built in to provide increased availability and fault-tolerance for the physical system.

3 Testbed Implementation

We envision (at least) three different realizations of the reference architecture: single simulation-based, federated simulation-based, and emulation- and implementation-based.

The *single simulation-based* instantiation has all elements implemented using a simulation framework and language, like Simulink/Stateflow from Mathworks [15]. We envision that the individual components of the architecture are implemented as Simulink subsystems that include the plant simulation, sensor simulations, simulations for the data acquisition and control activities on the RTUs, simulation of the computations performed on the SCADA servers, etc. For high-fidelity simulations we will model and simulate the implementation platforms as well: the OS schedulers and the networking mechanisms. The TrueTime toolsuite [23] provides a good example for doing this in the Simulink framework. For some, e.g. network attack scenarios these models will be extended to faithfully simulate the dynamic behavior of the network under attack.

The *federated simulation-based* instantiation uses several, dedicated, coordinated simulation engines that simulate the various architectural elements. Here, the key is that the individual simulation engines work with high-fidelity, industrial-grade models, possibly using off-the-shelf, commercial products. The same architectural elements are instantiated with a different technology, for example Speedup [2] for plant simulations, Omnet++ [19] for network simulation, and DEVS [24] for simulating software modules, etc. In this case the problem is the timed coordination across these simulation engines, but DoD's High-Level Architecture (HLA) [13] offers a platform to solve this problem. HLA provides services for simulation time coordination and data interchange during the simulation process, and several simulation engines have HLA interfaces implemented.

The emulation- and implementation-based instantiation uses actual commercial SCADA devices along with implementations of the software modules performing the data processing (running on realistic hardware), emulations of the network (running on a network emulator like EmuLab [9]), and real-time simulations for the plant (running on dedicated, high-performance hardware). We believe such an emulation/implementation-based realization is feasible and could be made highly realistic and scalable. Attacks on the network and computing nodes could be analyzed in a contained laboratory environment, which is safely decoupled from the 'real network', yet provides a highly realistic environment (e.g. like DETER [6] testbed).

4 Planned Experiments

SCADA networks are increasingly interconnected with other networks, and ensuring sufficient level of security for these networks is a challenge. An attack on any software component has an inevitable impact on the physical system with potential dire consequences. Therefore, securing both software and the physical system is essential. The security objectives that are of great importance in SCADA systems are integrity and availability. Integrity, in this framework, means that each component of the system functions and interacts with other components in the manner intended. This also includes the integrity of the collected data. The integrity directly maps into the reliability of the system.

In this work, we will implement specific experimental attack scenarios that compromise the integrity and availability of the entire system. Our goal is to develop methods to detect, predict and quantify the impact of these security attacks on the SCADA system.

An exhaustive analysis of all possible attacks is not feasible, but attacks trees are generally used in the literature to categorize different types of attacks [17]. In this work, we focus on specific scenarios and corresponding countermeasures, prioritizing threats that have a stronger impact on the integrity and availability of the entire system. The priority will be determined by the classification of vulnerabilities based on the consequences of the corresponding attack. The specific experiment scenarios that we analyze are:

- *Denial of service attacks on sensors:* We consider two types of denial of service attacks: jamming, and exploit of communication protocol design flaws. Jamming results in the loss of functionality by the network. TCP vulnerabilities or design flaws may also be leveraged. For example, a sensor node can be flooded with TCP requests which results in power exhaustion.

- *Integrity attacks:* Sensor outputs are essential to the situation awareness of a system. Consequently, sensors that transmit misleading outputs are a security threat. Our goal is to establish means to detect a sensor that emits corrupted data. In addition, we look at the software integrity of the RTU firmware to combat attacks that modify the behavior of the RTU. We consider software based attestation [20], secure code execution [21] and secure code update schemes for the RTUs [22].

- *Phishing attacks:* These are attacks against a web server that allows the attacker to access to protected information. This attack often is the first stage of a more complex attack [8].

In order to investigate these attacks, we need to provide the necessary modeling foundations on which threats and mitigation methodologies are based. We plan to develop mathematical and computational models for the interaction between the software infrastructure and the physical processes. The data-traffic generated by a SCADA system is complex and heterogeneous; the resources are dynamically distributed so that any analysis scheme has to adapt to continuous changes to the data-traffic patterns. In order to differentiate between normal changes and results of attacks or hardware failure, we plan to use accurate process modeling which is an abstraction of the time-evolution of the SCADA system.

5 Status

Work on the single simulation-based instantiation has started and we have a simulation of the physical infrastructure and its interaction with sensors and actuators. We are also working on a simple version of the emulation- and implementation-based instantiation of the testbed. We will use commercial RTUs and simulate the SCADA master server using commercial and custom applications. Our initial goal is to test and develop mechanisms to ensure the integrity of the RTUs.

6 Next Steps

In the following months we plan to improve upon our single simulation-based instantiation and simulate the SCADA servers, RTUs and sensors as well. We will then test high-level attack senarios and solutions on this testbed. The results of these tests will be used generate an attack tree to categorize attack senarios and countermeasures. We eventually plan to shift our single simulation-based instatiation to a federated simulation-based instantiation of the testbed. This testbed will allow us to test various attack senarios and solutions in a realistic but simulated envrionment. We will also continue improving our emulation- and implementation-based instantiation along the way to allow for tests on a more realistic and scalable environment.

7 Conclusion

It is imperative that SCADA systems be secured, given their critical nature. The SCADA testbed will help us design and test solutions to various attacks against SCADA systems. We hope to design retrofit solutions that will help secure existing and legacy SCADA systems as well as cutting-edge solutions that will help protect future SCADA systems for many years to come.

8 Acknowledgements

This work was supported in part by TRUST (Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: AFOSR (#FA9550-06-1-0244), BT, Cisco, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia, and United Technologies.

References

[1] 21 Steps to Improve Cyber Security of SCADA Networks. U.S. Department of Energy white paper, 2005.

[2] Aspentech. http://www.aspentech.com/

[3] The C2 Wind Tunnel, https://wiki.isis.vanderbilt.edu/c2w/

[4] The Center for SCADA Security. Sandia National Labrotories, http://www.sandia.gov/scada/testbeds.htm.

[5] Henrik Christiansson and Eric Luiijf. Creating a European SCADA Security Testbed. In *IFIP International Federation for Information Processing*, Springer Boston 2007.

[6] The DETER Testbed, http://www.deterlab.net/

[7] DNP. http://www.dnp.org/

[8] G. Dondossola, J. Szanto, M. Masera, I. Nai Fovino. Evaluation of the effects of intentional threats to power substation control systems. In *Proceedings of the International Workshop on Complex Network and Critical Infrastructure Protection*, 2006.

[9] Emulab. http://www.emulab.net/

[10] Scott Fluhrer and Itsik Mantin and Adi Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. *Lecture Notes in Computer Science, 2259*,2001.

[11] HART Communication Foundation. *WirelessHART Technical Data Sheet*, 2007. www.hartcomm.org

[12] Carl Hartung, James Balasalle and Richard Han. Node Compromise in Sensor Networks: The Need for Secure Systems. *Department of Computer Science University of Colorado at Boulder*, 2005

[13] High-Level Architecture, IEEE Standard 1516. www.ieee.org

[14] Vinay M. Igure, Sean A. Laughter and Ronald D. Williams, Security issues in SCADA networks. In *Computers & SecurityVolume 25, Issue 7*, October 2006, Pages 498-506.

[15] MathWorks Simulink. http://www.mathworks.com

[16] Modbus-IDA. http://www.modbus.org/

[17] A. Moore, R. Ellison and R. Linger. Attack modelling for information security and survivability. In *SEI*, 2001.

[18] Hamed Okhravi, Chris Grier, Matt Davis, Zeb Tate, David Nicol, and Tom Overbye. Cyber-Security Simulation Testbed. http://www.iti.uiuc.edu/tcip/tcip_presentations.html

[19] Omnet++. http://www.omnetpp.org/

[20] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla. SWATT: SoftWare-based ATTestation for Embedded Devices. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, May 2004.

[21] A. Seshadri, M. Luk, E. Shi, A. Perrig, L. van Doorn, and P. Khosla. Pioneer: Verifying Integrity and Guaranteeing Execution of Code on Legacy Platforms. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, Brighton, United Kingdom, October 2005.

[22] A. Seshadri, M. Luk, A. Perrig, L. van Doorn, and P. Khosla. SCUBA: Secure Code Update By Attestation in Sensor Networks. In *ACM Workshop on Wireless Security (WiSe 2006)*, Los Angeles, CA, September 29, 2006.

[23] TrueTime. http://www.control.lth.se/truetime/

[24] Bernard Zeigler, Tag Gon Kim, Herbert Praehofer (2000). *Theory of Modeling and Simulation*, Second Edition, Academic Press, New York. ISBN 978-0127784557.