Simulation of Network Security Attacks on SCADA Systems

Rohan Chabukswar
Carnegie Mellon University

Bruno Sinópoli
Carnegie Mellon University

Gabor Karsai
Vanderbilt University

Annarita Giani
University of California Berkeley

Himanshu Neema
Vanderbilt University

Andrew Davis
Vanderbilt University

Abstract—**Security is a major issue affecting SCADA systems, designed and deployed in the last decade. Simulation of network attacks on a SCADA system presents certain challenges, since even a simple SCADA system is composed of models in heterogenous domains and simulation environments. Here, we simulate a chemical plant and its controller, communicating through an ethernet network. We simulate and observe the effects of a common network attack on the availability of the system.**

I. Introduction

Supervisory Control And Data Acquisition (SCADA) systems are computer-based monitoring tools that are used to manage and control critical infrastructure functions in real time, like gas utilities, power plants, chemical plants, traffic control systems, etc. A typical SCADA system would consist of a SCADA Master which provides overall monitoring and control for the system, local process controllers called Remote Terminal Units (RTUs), sensors and actuators and a network which provides the communication between the Master and the RTUs.

A. Security of SCADA Systems

SCADA systems are designed for a long life span, usually in decades. The SCADA systems currently installed were designed at a time when security was not paramount, which is not the case today. Furthermore, SCADA systems are now connected to the internet for remote monitoring and control. Due to this, the systems are susceptible to network security problems which arise through such a connection.

Despite of these evident security risks, SCADA systems are cumbersome to upgrade for several reasons. Firstly, adding security features often implies a large downtime, which is not desirable in systems like power plants and traffic control systems. Secondly, SCADA devices with embedded codes would have to be completely replaced to add new security protocols. Thirdly, the networks used in a SCADA system are usually customized for that system and cannot be generalized.

Security of legacy SCADA systems and design of future systems thus rely heavily on the assessment and rectification of security vulnerabilities of SCADA implementations in realistic settings.

B. Simulation of SCADA Systems

Even a simple SCADA system is composed of several units in various domains like dynamic systems, networks and physical environments, and each of these units can be modeled using a variety of available simulators and/or emulators. An example system could include simulating controller and plant dynamics in Simulink or Matlab, a three dimensional physical environment in Delta3D, network architecture and behavior in a network simulator like OMNeT++ and maybe even human organization and co-ordination in Colored Petri Nets (CPN). An adequate simulation of such a system necessitates the use of an underlying software infrastructure that connects and relates the heterogeneous simulators in a logically and temporally coherent framework.

II. C2WindTunnel

One infrastructure suitable for such an application is the C2WindTunnel. It is a software suite which enables various simulation engines to interact and transmit data to and from one another, and log and analyze the real time simulation results. C2WindTunnel is based on the DoD/HLA run-time integration platform.

A. High Level Architecture (HLA)

High Level Architecture (HLA), defined under IEEE standard 1516, is the standard for ensuring interoperability and reusability of models and simulation components. The primary goal of HLA is to provide a general purpose infrastructure for distributed simulation and analysis by establishing interoperability on a technical, syntactic and semantic levels. Prior to it becoming an IEEE standard, it was a policy in the US Department of Defense for all models and simulations to comply with the standard.

An HLA compliant simulation is called a Federate, and a system of such simulations connected via the RTI is called a Federation.

B. Run-Time Infrastructure (RTI)

The software implementation of HLA is called a Run-Time Infrastructure (RTI). There are several commercial and open-source RTIs available in the market, some of which have been verified by the US Defense Modeling and Simulation Office.

The RTI is basically a collection of software which provides a set of commonly required services, described by the HLA Interface Specification, to multiple simulation systems. Apart from federation and object management, RTI co-ordinates the exchange of interacting events and data among the federates in a system. The time management services provided by the RTI ensure advancement of the simulation time in an orderly fashion among all the federates.

Initially, the RTI sets the logical time and rate of time advance for the federation. The federate manager uses HLA-specified synchronization points to guarantee that all federates are ready to proceed with the simulation. Only when each federate has reported readiness to proceed with the simulation, does the federate manager allow all federates to commence the simulation. Each federate individually runs its simulation engine for one time step (T) of the logical time. Each federate then communicates to the RTI the data it needs to send to the rest of the federates during this time T as well as in a certain preset lookahead time (L) further. Then, each federate (again, individually) requests permission from the RTI to run its simulation engine for the next time step. The RTI checks what data that federate will receive from the rest of the federation during the future T time period. To accomplish this, the RTI might need to wait till all the federates have finished executing till atleast $(L - T)$ time steps behind the current logical time of the concerned federate. When the RTI has this information, it conveys the data to the federate, along with the permission to simulate for the next time step T. The system simulation proceeds step-by-step in this way for the remainder of the simulation.

Data communication between the federates is handled using a publish and subscribe mechanism. The output binding of each federate publishes all interactions onto a common HLA bus. The input binding subscribes to the events and objects that are of interest to the federate.

C. C2WindTunnel Modeling Environment

Based on the C2WindTunnel models, data flow, timing and parameters configuration files are generated for the various simulation components of the system, which determine how each component is connected to the simulation. Domain specific data models are used to transform data from one simulation model to another.

Figure 1 shows the structure of a simulation undertaken using C2WindTunnel

III. Demonstration

In this section, we demonstrate the effects of certain security attacks on a simulated SCADA system.

The system is a chemical plant and its controller, and data is sent to and from the plant using an ethernet network. The chemical plant used is a simplified version of the famous Tennessee Eastman challenge problem, with eight states, four manipulated variables and ten outputs. The process consists of a single vessel of fixed volume, in which a single, irreversible reaction occurs in the vapor phase. The two reactants are non-condensable, and the product is a non-volatile liquid. There are two feeds into the vessel, the first delivers both the reactants and minute amounts of an inert gas. The second feed delivers one of the reactants in its pure form, to compensate for the ratio of the reactants in the first feed. The product and purge rates are adjusted by the liquid inventory and vessel pressure respectively, which are two of the manipulated variables, along with the two independent feed rates.

The control of the plant involves regulating the product rate to a specified value by manipulating the flows in the two feeds and the purge. There are other constraints on the system, including maximum operating pressure and saturation constraints on the flows.

This problem and its controller was proposed by N. Lawrence Ricker in [1]. Ricker also provides the simulation for the plant as a Simulink S-Function. The robust model-predictive controller is one of several proposed by Ricker. It uses only four out of the ten available sensor outputs, and controls all four manipulated variables. This was separately implemented in Simulink.

To complete the system, an ethernet network was added for communication between the plant and its controller. The network was designed to be a realistic implementation of one in a chemical plant, where a single router would collect data from plant sensors which are physically close to it (and to each other). The network model is simulated in OMNeT++, a generic discrete event simulation package using INET network protocols.

For the abstract, a simpler network is implemented, where only one router collects data from all of the sensors.

A. Model Integration

1) Simulink: To the original models of the plant and the controller, input and output bindings were added which determine the signal flow across the simulation through the RTI. The `.m` code for the receiver and sender bindings was generated by the integration model defined in GME (Generic Modeling Environment), a configurable toolkit for creating domain-specific modeling and program synthesis environments. The GME integration model also generated the Java code which represents the Simulink federate. This federate handles the runtime communication to the RTI.

2) OMNeT++: Apart from the above steps, the integration of the OMNeT++ model presented certain challenges. The time management of RTI had to replace OMNeT++'s scheduler. Also, scalability had to be provided for, to avoid overloading the RTI bus yet still capture interesting behaviours in the network.

B. Attacks

The current attack simulated on the network is a common distributed denial of service (DDoS) attack. In such an attack, the target is saturated with external communication requests so that it cannot handle the legitimate traffic of the system, or atleast, is rendered so slow in handling the traffic, that it

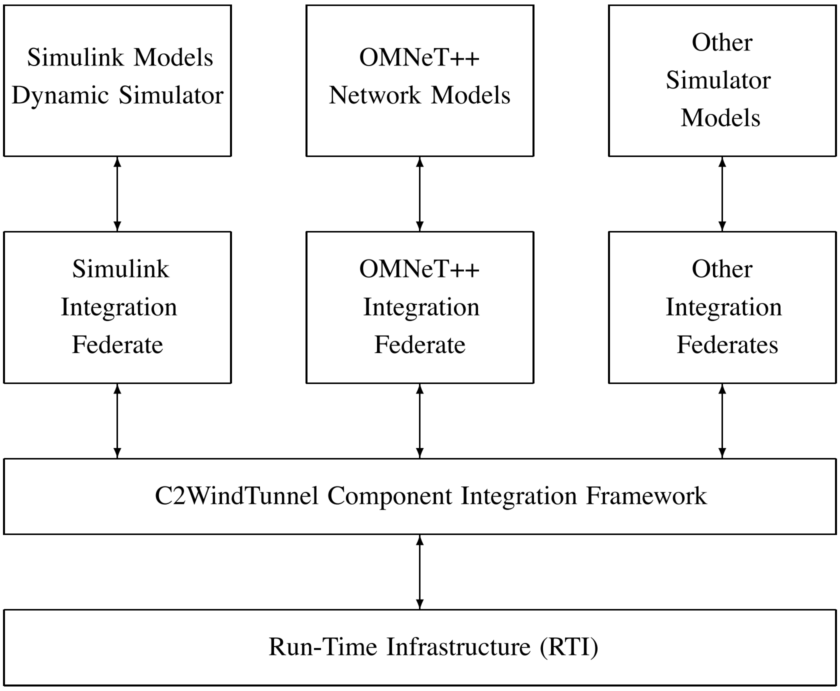

Fig. 1. C2WindTunnel Simulation: The different models are domain specific, and the federates are reusable C2WindTunnel simulators.

is effectively unavailable for legitimate transfer of data. The targets, durations and the number of attacks in the simulation can be specified beforehand.

IV. Observations

When even a single one of the routers is under a full-fledged DDoS attack, the network is essentially broken at that point. The controller will be rendered blind to sensors that the router collects data from. If the router is also involved in the transfer of data from the controller to the plant, the plant will be rendered unresponsive to the commands given by the controller. This will result in a loss of the regulatory function of the controller, which can potentially cause a variety of damage to the plant, from an unwanted change in the operating cost, to physical damage to plant equipment.

For the simpler network model, the similar disruption causes complete loss of communication between the plant and controller. The plant settles into an uncontrolled state for the duration of the attack, from which it can only recover and resume normal operation after the attack has ceased. This effect can be observed by monitoring the operating cost during the simulation, as seen in Figure 2.

V. Conclusion

A familiar security attack was simulated on a SCADA system, and reasonable effects of the attacks were observed in the functioning of the system. The chemical plant was a proof-of-concept implementation of a system composed of models in different domains and simulation environments. The use of C2WindTunnel facilitated the interaction and data transfer between the environments.

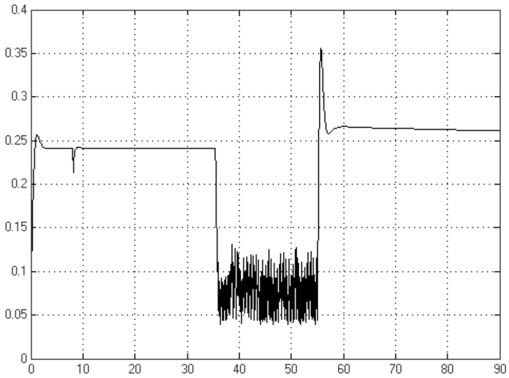

Fig. 2. Effect of DDoS attack on instantaneous operating cost

VI. Future Work

The Distributed Denial of Service attack is one attack on the availability of a system. Future work involves observing the effect of other common network security attacks on integrity and confidentiality of the data as well, like eavesdropping, misdirection and spoofing.

Another direction for future work involves simulation of systems including hardware-in-the-loop.

References

[1] N. Lawrence Ricker, *Model predictive control of a continuous, nonlinear, two-phase reactor*. Journal of Process Control, Volume 3, Issue 2, May 1993, Pages 109-123.

[2] J. O. Calvin, R. Weatherly, *An introduction to the high level architecture (HLA) runtime infrastructure (RTI)*. Proceedings of the 14th Workshop on Standards for the Interoperability of Defence Simulations, Orlando, FL, March 1996, pp. 705-715.