INT J COMPUT COMMUN, ISSN 1841-9836
Vol.7 (2012), No. 4 (November), pp. 674-687

Impact of Network Infrastructure Parameters to the Effectiveness of Cyber Attacks Against Industrial Control Systems

B. Genge, C. Siaterlis, M. Hohenadel

Béla Genge, Christos Siaterlis and Marc Hohenadel
Institute for the Protection and Security of the Citizen
European Commission, Joint Research Centre
Via E. Fermi, 21027, Ispra (VA), Italy
E-mail: {bela.genge, christos.siaterlis, marc.hohenadel}@jrc.ec.europa.eu

Abstract:
The fact that modern Networked Industrial Control Systems (NICS) depend on Information and Communication Technologies (ICT), is well known. Although many studies have focused on the security of SCADA systems, today we still lack the proper understanding of the effects that cyber attacks have on NICS. In this paper we identify the communication and control logic implementation parameters that influence the outcome of attacks against NICS and that could be used as effective measures for increasing the resilience of industrial installations. The implemented scenario involves a powerful attacker that is able to send legitimate Modbus packets/commands to control hardware in order to bring the physical process into a critical state, i.e. dangerous, or more generally unwanted state of the system. The analysis uses a Boiling Water Power Plant to show that the outcome of cyber attacks is influenced by network delays, packet losses, background traffic and control logic scheduling time. The main goal of this paper is to start an exploration of cyber-physical effects in particular scenarios. This study is the first of its kind to analyze cyber-physical systems and provides insight to the way that the cyber realm affects the physical realm.
Keywords: cyber attacks, Industrial Control Systems, SCADA, security.

1 Introduction

Modern Critical Infrastructures (CI), e.g. power plants, water plants and smart grids, rely on Information and Communication Technologies (ICT) for their operation since ICT can lead to cost optimization as well as greater efficiency, flexibility and interoperability between components. In the past CIs were isolated environments and used proprietary hardware and protocols, limiting thus the threats that could affect them. Nowadays CIs, or more specifically Networked Industrial Control Systems (NICS), are exposed to significant cyber-threats; a fact that has been highlighted by many studies on the security of Supervisory Control And Data Acquisition (SCADA) systems [1, 2].

The recently reported Stuxnet worm [3] is the first malware specifically designed to attack NICS. Its ability to reprogram the logic of control hardware in order to alter physical processes demonstrated how powerful such threats can be. Stuxnet was a concrete proof of a successful cyber-physical attack but by no means a trivial attack. It required a thorough knowledge of the physical system, software and OS vulnerabilities. In this paper we consider an adversary with a lower level of sophistication that instead of reprogramming the highly specialized hardware (PLCs) as in the Stuxnet case, he exploits the ability of control hardware to communicate with remote stations using well-established protocols such as TCP, that are normally used by the operator to read sensor values, e.g. pressure, temperature, and control actuators, e.g. valves. Furthermore, we identify the communication and control logic implementation parameters that influence the outcome of cyber attacks against NICS and that could be used as effective measures for increasing the resilience of industrial installations. In our study the attacker is located outside the plant, somewhere in the Internet, from where he/she exploits the capability of control hardware to communicate with remote stations. Although direct connections between control hardware and the Internet are usually avoided in NICS implementations [5], the attacker might install a malware on a station with an Internet connection located within the corporate network that could be used to forward messages to the control hardware. We consider that the control hardware is running legitimate code designed to keep the physical process in a normal operating point, while the attacker tries to bring it to a *critical state*, i.e. dangerous, or more generally unwanted state of the system [6]. The method employed by the attacker is a repeated transmission of commands to the control hardware that open/close specific valves. The main contribution of this paper is that it is the first of its kind to analyze cyber-physical systems and provides insight to the way that the cyber realm affects the physical realm.

The real applicability of the implemented scenario is confirmed not only by Stuxnet, but by other past events as well. One example in this sense is a 2002 penetration test done by a security firm for a power company located in California. The testers parked their van outside a remote substation, where they noticed a wireless antenna. Without leaving their vehicle they managed to connect to the system and within 20 minutes they have not only mapped the entire network, but also "they were talking to the business network and had pulled off several business reports" [4]. By taking these events and adapting them to our scenario we can further imagine that the testers are in fact attackers that install a remotely accessible gateway and launch their attack from the anonymity provided by the Internet.

The attack scenario has been implemented with the help of our previously developed framework [7] that uses simulation for the physical components and an emulation testbed based on Emulab [8, 9] to recreate the cyber part of NICS, e.g., SCADA servers, corporate network, etc. In the implemented scenario we have used the model of a Boiling Water Power Plant developed by Bell and Åström [10].

The paper is structured as follows. After an analysis of related work in Section 2 we present a brief overview of the experimentation framework in Section 3. The methodology and implemented experiments are presented in Section 4 and 5, respectively. The paper concludes in Section 6.

2 Related Work

An approach where real sensors and actuators, combined with simulated PLCs and communication protocols were used to study cyber-physical systems has been proposed by Queiroz, *et al.* [15]. Their study showed that while PLCs are under a DoS attack, operators might take delayed or wrong decisions that could disrupt the operation of the plant. A similar experiment has also been documented by Davis, *et al.* [16] that used the PowerWorld server to study the effects of communication delays between the physical process and human operators. In the same direction, the work of Chabukswar, *et al.* [17] proved that a DDoS attack against communication nodes between controllers and sensors causes the PLCs to take wrong decisions based on old sensor values. Cárdenas, *et al.* [18], went further by not only documenting the effects of DoS attacks on sensors, but also proposing a new detection mechanism and possible countermeasures.

The previously mentioned approaches have demonstrated the effectiveness of DoS attacks, but without reaching a sophistication level that would have allowed the attacker to reprogram the low level control logic of the PLCs. This fact sets an important barrier in terms of knowledge, skills and efforts required by the attacker, as was the case of Stuxnet, where developers had also knowledge of the PLC code, OS and hardware details. In this category we find the work of Nai Fovino, *et al.* [5] that have proposed an experimental platform for studying the effects of cyber attacks against NICS. In their paper the authors describe several attack scenarios, including DoS attacks and worm infections that send Modbus packets to control hardware. Although the authors provide a wide set of countermeasures, they do not identify communication parameters that affect the outcome of the attacks. Moreover, in our analysis we have also identified installation-specific parameters that can directly affect the resilience of physical processes.

3 Framework Overview

After providing a brief description of a typical NICS architecture, this section presents a short overview of our previously developed experimentation framework used in our experiments.

3.1 Process Control Architecture Overview

In modern NICS architectures, one can identify two different control layers: (i) the physical layer composed of all the actuators, sensors, and generally speaking hardware devices that physically perform the actions on the system, e.g. open a valve, measure the voltage in a cable; (ii) the cyber layer composed of all the ICT devices and software which acquire the data, elaborate low level process strategies and deliver the commands to the physical layer. The cyber layer typically uses SCADA protocols to control and manage the physical devices within the cyber layer. The ”distributed control system” of the cyber layer is typically split among two networks: the *control network* and the *process network*. The process network usually hosts all the SCADA (also known as SCADA Masters) and HMI (Human Machine Interface) servers. The control network hosts all the devices which, on the one side control the actuators and sensors of the physical layer and on the other side provide the ”control interface” to the process network. A typical control network is composed of a mesh of PLCs (Programmable Logic Controller). From an operational point of view, PLCs receive data from the physical layer, elaborate a ”local actuation strategy”, and send back commands to the actuators. PLCs execute also the commands that they receive from the SCADA servers (Masters) and additionally provide, whenever requested, detailed physical layer data.

3.2 Framework Architecture

The previously developed framework [7] follows a hybrid approach, where the Emulab-based testbed recreates the control and process network of NICS, including PLCs and SCADA servers, and a software simulation reproduces the physical processes. The architecture, as shown in Figure 1, clearly distinguishes 3 layers: the cyber layer, the physical layer and a link layer in between. The cyber layer includes regular ICT components used in SCADA systems, while the physical layer provides the simulation of physical devices. The link layer, i.e. cyber-physical layer, provides the ”glue” between the two layers through the use of a shared memory region.

The physical layer is recreated through a soft real-time simulator that runs within the SC (Simulation Core) unit and executes a model of the physical system. The simulator’s execution time is strongly coupled to the timing service of the underlying operating system (OS). Throughout the paper the term *time step* is used to denote the time between two successive executions of the physical model in the simulator. The cyber layer is recreated by an emulation testbed that uses the Emulab architecture and software [8,9] to automatically and dynamically map physical components, e.g. servers, switches to a virtual topology. Besides the process network, the cyber layer also includes the control logic code that in the real world is run by PLCs. The control code can be run sequentially or in parallel to the physical model. In the sequential case, a *tightly coupled* code (TCC) is used, i.e. code that is running in the same memory space with the model, within the SC unit. In the parallel case a *loosely coupled* code (LCC) is used, i.e. code that is running in another address space, possibly on another host, within the R-PLC unit (Remote

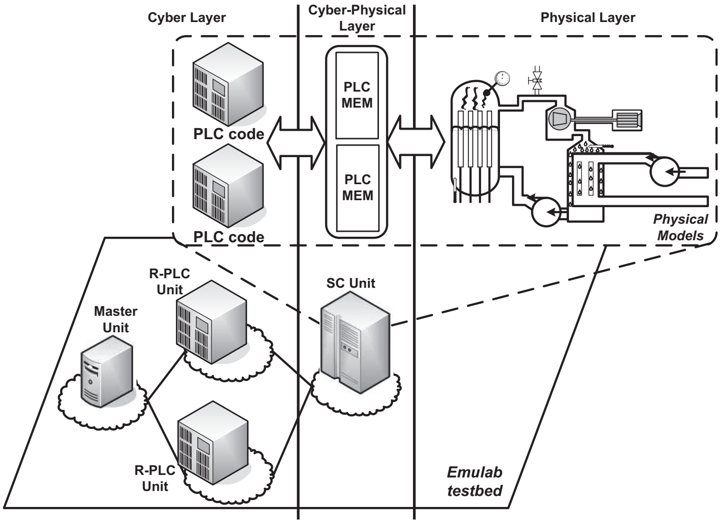

Figure 1: Experimentation framework architectural overview

PLC). The unit that implements global decision algorithms based on the sensor values received from the R-PLC units is also present in the proposed framework as the *Master* unit. The cyber-physical layer incorporates the PLC memory, seen as a set of registers typical of PLCs, and the communication interfaces that "glue" together the other two layers. Prototypes of SC, R-PLC and Master Units have been developed in C# (Windows) and have been ported and tested on Unix-based systems (FreeBSD, Fedora and Ubuntu) with the use of the *Mono* platform. Matlab Simulink was used as the physical process simulator (physical layer). From Simulink models the corresponding 'C' code is generated using Matlab RTW. The communication between SC and R-PLC units is handled by .NET's binary implementation of RPC (called *remoting*) over TCP. For the communication between the R-PLC and Master units, we used the Modbus over TCP protocol.

4 Methodology

In this section we provide a description of the methodology we used, including a description of the scenario and experimental setup.

4.1 Scenario

Previous security events [3, 4] involving NICS showed that attackers can (easily) compromise stations located within an installation's internal network. These stations can then be used as gateways for downloading malicious code and for remotely controlling other stations, including control hardware such as PLCs. The implemented scenario assumes that there is already a compromised station providing access to PLCs. This is used by the attacker to send remote Modbus commands in order to bring the physical process into a critical state.

The physical process used in our experiments is the Boiling Water Power Plant (BWPP) model developed by Bell and Åström [10]. Within this context the critical state is given by an increased steam pressure that is more than twice the value of a typical operating point. Although to the best of our knowledge the literature does not mention anything about the consequences of running the process with these parameters, we assumed that this might cause physical damages and therefore could become a desired target for the attacker. Furthermore, we assumed that the attack is conducted for a limited time period of 10 minutes, as immediately after it is started the process experiences significant deviations from its normal operating point. Consequently, alarms might be turned ON and human operators might intervene by switching OFF devices or disconnecting equipment. As shown later in this paper, the 10 minute time period is more than enough for the attacker to bring the physical process into a critical state.

Our experimental results presented in the next section show that the outcome of the attack is affected not only by network delays, packet losses and background traffic, but also by the execution of PLC control code and the speed of control valves. These can be used as effective measures to increase the resilience of physical processes confronted with cyber attacks.

4.2 Experimental Setup

The previously described scenario has been implemented in the Joint Research Centre's (JRC) Experimental Platform for Internet Contingencies (EPIC) laboratory. The Emulab testbed included nodes with the following configuration: FreeBSD OS 8, AMD Athlon Dual Core CPU at 2.3GHz and 4GB of RAM. We have emulated network delays, packet losses and background traffic with the *Dummynet* [11, 12] and *iperf* [13] software in order to recreate a dynamic and unpredictable environment such as the Internet.

As shown in Figure 2, the experimental setup consisted of 6 hosts: 1 host for running the SC unit, 3 hosts for running the R-PLC units, 1 host for running the Compromised Station and 1 host for the attacker. The attacker uses the Compromised Station to forward Modbus packets to R-PLC units and finally to write the PLC memory within the SC unit. The control code that is in charge of maintaining the BWPP at a constant operating point has been implemented as TCC code, where *TCC1* controls the fuel valve, *TCC2* controls the steam valve and *TCC3* controls the feed-water valve. The role of the R-PLC units is simply to enable access to the physical model using the Modbus/TCP protocol. Although R-PLC units can also run control code, the decision to implement control code as TCCs has been made based on the granularity of the process model execution step that needs to be less than 1ms. This is needed in order to emulate PLC *tasks* that can be scheduled to run within milliseconds. Consequently, the chosen time step for the physical model was of 0.5ms.

Network delays, background traffic and packet losses, specific to a dynamic environment such as the Internet, have been emulated between the attacker's station and the Compromised Station. The limited bandwidth and communication capabilities of PLCs have been emulated with 10Mbit/s Lans (*Lan2* and *Lan1*, respectively), while using a 100Mbit/s Lan (*Lan0*) between R-PLCs and the SC unit in order to maximize the performance of the interaction between R-PLC units and the BWPP model.

4.3 Boiling Water Power Plant

As already mentioned, in this paper we use the Boiling Water Power Plant (BWPP) model developed by Bell and Åström in [10]. It models a 160MW oil-fired electric power plant based on the Sydsvenska Kraft AB plant in Malmö, Sweden. The operation of the process is controlled by three valves, i.e. fuel valve, steam valve and feed-water valve, while the operator is able to monitor the process by reading three sensors: steam pressure, water level and generated electricity. The following equations describe the dynamics of the physical process [10]:

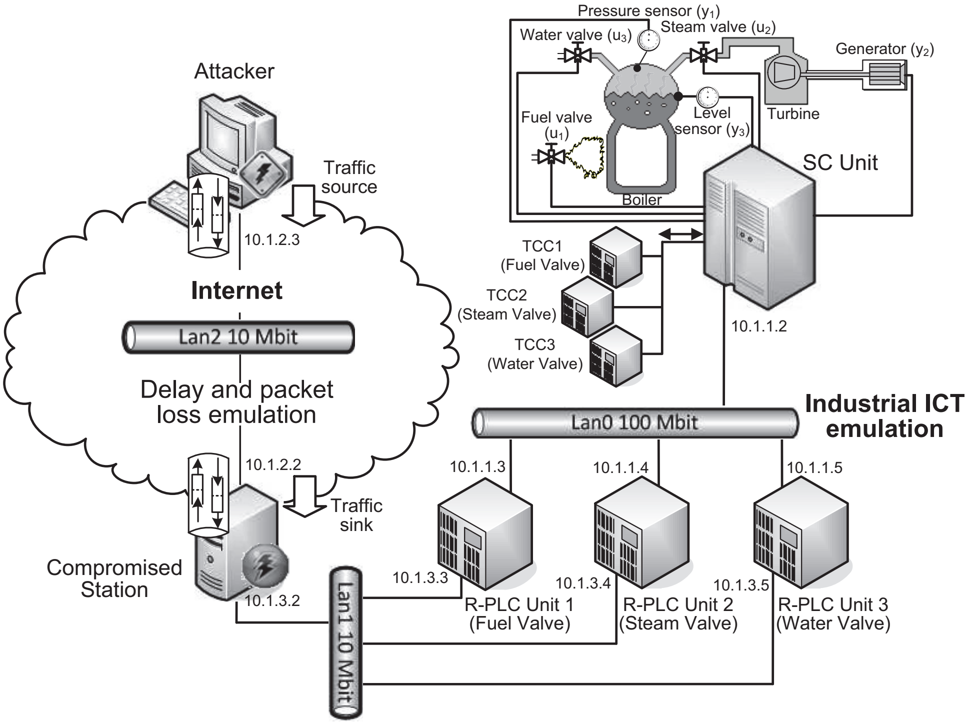

Figure 2: Experimental setup

$$
\begin{aligned}
\dot{x}_1 &= -0.0018u_2x_1^{9/8} + 0.9u_1 - 0.15u_3,\\
\dot{x}_2 &= (0.073u_2 - 0.016)x_1^{9/8} - 0.1x_2,\\
\dot{x}_3 &= (141u_3 - (1.1u_2 - 0.19)x_1)/85,\\
y_1 &= x_1,\\
y_2 &= x_2,\\
y_3 &= 0.05(0.1307x_3 + 100s_q + e_r/9 - 67.975),\\
s_q &= \tfrac{(1-0.001538x_3)(0.8x_1-25.6)}{x_3(1.0394-0.0012304x_1)},\\
e_r &= (0.85u_2 - 0.147)x_1 + 45.59u_1 - 2.514u_3 - 2.096,\\
0 &\leq u_i \leq 1 (i = 1, 2, 3),\\
\dot{u}_1 &\leq 0.007/sec, -1.0 \leq \dot{u}_2 \leq 0.1/sec, \dot{u}_3 \leq 0.05/sec,
\end{aligned}
\tag{1}
$$

where x_1, x_2 and x_3 denote the steam pressure (kg/cm^2), electric output (MW), and fluid density (kg/m^3), respectively; y_1, y_2 and y_3 denote the outputs of the model, where y_3 is the water level deviation (m); u_1, u_2 and u_3 are the valve positions for fuel flow, steam flow and feed-water flow, respectively; s_q and e_r denote the steam quality and evaporation rate (kg/s), respectively.

In our experiments we used a normal operating point for the BWPP in which the pressure equals 108 kg/cm^2, that was achieved by keeping the three valves in the following *normal valve positions* (NVP): $u_1 = 0.34$, $u_2 = 0.69$ and $u_3 = 0.433$ [14]. As a consequence, TCCs, that implement the control logic code, must maintain constant the position of the three valves in order to provide a constant steam pressure of 108 kg/cm^2. For the critical state we consider a pressure of 250 kg/cm^2 that is more than twice the value of the steam pressure of the process running in the previously mentioned normal operating point. The attacker is able to bring the

BWPP into the critical state by continuously sending Modbus packets (around 100/sec for each valve) that keep the steam and feed-water valves in the CLOSED position ($u_2 = 0$ and $u_3 = 0$, respectively) and the fuel valve in the OPENED position ($u_1 = 1$). Throughout this paper we use the term *attacker's valve positions* (AVP) to denote the position of the three valves in the attacker's setting.

5 Attacks and Analysis

In this section we analyze the influence of several parameters on the cyber attacks launched remotely from the Internet, as described in the scenario from the previous section. The analysis is conducted in two phases: in the first phase we analyze the ability of the attacker to maintain the control valves in the AVP; in the second phase we analyze the ability of the attacker to increase the steam pressure to 250 kg/cm^2, thus actually bringing the plant into the critical state. While the goal of the second phase is clear, the goal of the first one needs further explanation. The actual goal of the first phase is to analyze the reaction of PLCs in terms of commands sent to the valves, reaction that might provide assistance in the rationale of the results in the second phase. In both phases we have measured the influence of: PLC task scheduling (TS) every 100ms and 1ms; network delays of 0s, 0.5s and 3s; packet losses of 1%, 5% and 10%; and background traffic of 2.5Mbit/s, 5Mbit/s and 10Mbit/s. Such extreme values for network delays, i.e. 3s, and packet losses, i.e. 10%, can rarely be measured over the Internet (possibly over satellite links or multiple intermediate proxies). Nevertheless, we have included them in our analysis in order to justify our statements related to the required magnitude of these parameters for influencing the outcome of the attack. For each configuration setting, representing a combination of PLC TS, network delays, packet losses and background traffic we have run one experiment for 10 minutes. In total, we have run 540 experiments in 9 hours.

5.1 The Effect of the Cyber Attack on the Position of Control Valves

The implementation of Modbus over TCP allows attackers to remotely control the three valves, i.e. fuel valve, steam valve, feed-water valve, thus providing a certain anonymity to the attacker. Nevertheless, this status is compensated by fluctuating parameters such as network delays, packet losses and background traffic that can have a major effect on the outcome of the attack. In this sub-section we analyze the effects of these parameters and two different TS, i.e. 100ms and 1ms, on the position of the three control valves.

100ms Task Scheduling

The effect of network delays on the position of control valves for a 100ms TS time is given in Figure 3. Each sub-figure shows the position of a specific control valve during the 10 minute attack, starting with the NVP shown at $t = 0$ minutes and following with the changes induced by the attacker that, as shown by most of the figures, bring the valves into the AVP. The differences we observe in the behavior of the three valves are caused by the motion speed of each valve that is different in each case (Equation 1). Thus, the fastest to open and close is the steam valve (0.1/sec for opening and 1/sec for closing) followed by the feed-water valve (0.05/sec) and by the fuel valve (0.007/sec). Consequently, the greater the speed, the higher the fluctuations that we see in Figure 3 and the higher the ability of PLCs to maintain the NVP.

One of the main conclusions from our results (Figure 3) is that network delays are beneficial to the physical process when confronted with cyber attacks. Nevertheless, the attacker is able to bring the valves into the AVP even for network delays of 3s. The attacker is most successful with

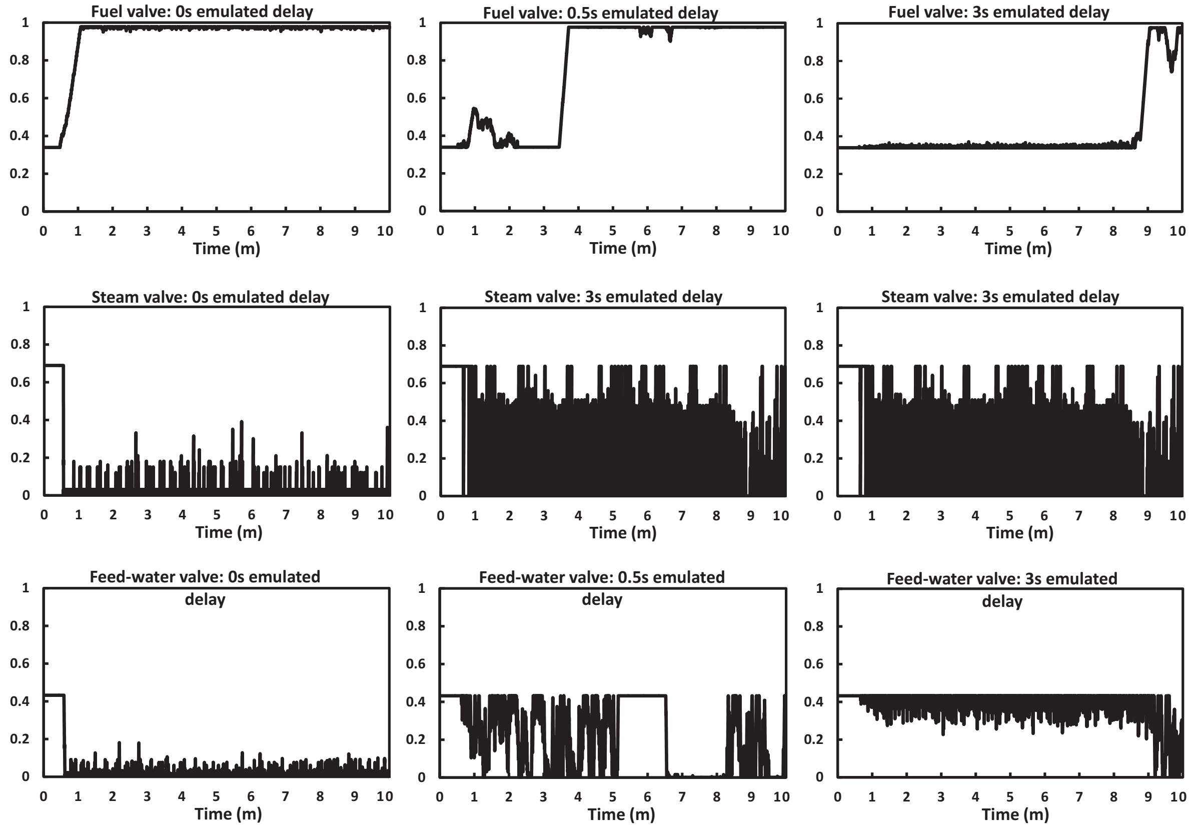

Figure 3: Effect of network delays on control valve positions for 100ms TS, 1% loss rate and 2.5Mbit/s background traffic

the fuel valve as this has the lowest speed and the control code is running only once every 100ms, while the attacker sends one Modbus command every 10ms, i.e. 100/sec. For the other two valves the PLCs are able to produce a slight deviation from the AVP; however, these are unable to bring the valves back to the NVP. As we increase the network delays we notice that PLCs are causing larger deviations from the AVP and in the extreme case of 3s the attacker is able to produce only small deviations from the NVP for the fuel and feed-water valves. Nevertheless, the attacker is still able to close the steam valve, as its closing speed (1/sec) exceeds 10 times its opening speed (0.1/sec). Based on these results we can conclude that only extreme network delays, e.g. 3s, have a major influence on the outcome of the attack. In addition, an attacker could successfully exploit a different opening and closing speed of valves, that could be interpreted as a slower reaction of the PLCs.

Going further, in Figure 4 we show the effect of packet losses on the position of the valves for the same TS of 100ms. By increasing the packet loss from 1% (Figure 3) to 5% and 10% (Figure 4) we also increase the deviations from AVP. Nevertheless, even with higher packet losses PLCs are unable to maintain the NVP, not even in the case of slower valves such as the fuel and feed-water valves. As shown in the same figure, we have also experimented with extreme packet losses of 10%. However, as the attacker uses a 10 times higher packet rate than the control code scheduling rate, even in this case PLCs are unable to keep the valves in the NVP.

We have also investigated the effect of background traffic on the three valves, shown in Figure 5. By increasing the background traffic from 2.5Mbit/s (Figure 3) to 5Mbit/s (Figure 5) we do not notice major effects, as the maximum network capacity is 10Mbit/s and the 100 Modbus

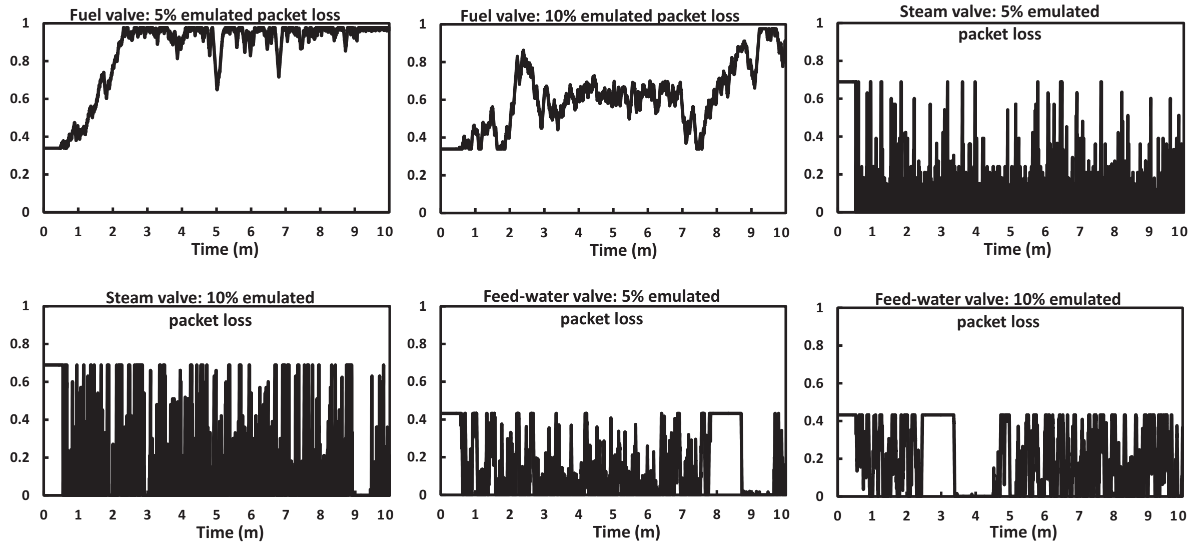

Figure 4: Effect of packet loss on plant valve positions for 100ms TS, 0s emulated delay and 2.5Mbit/s background traffic

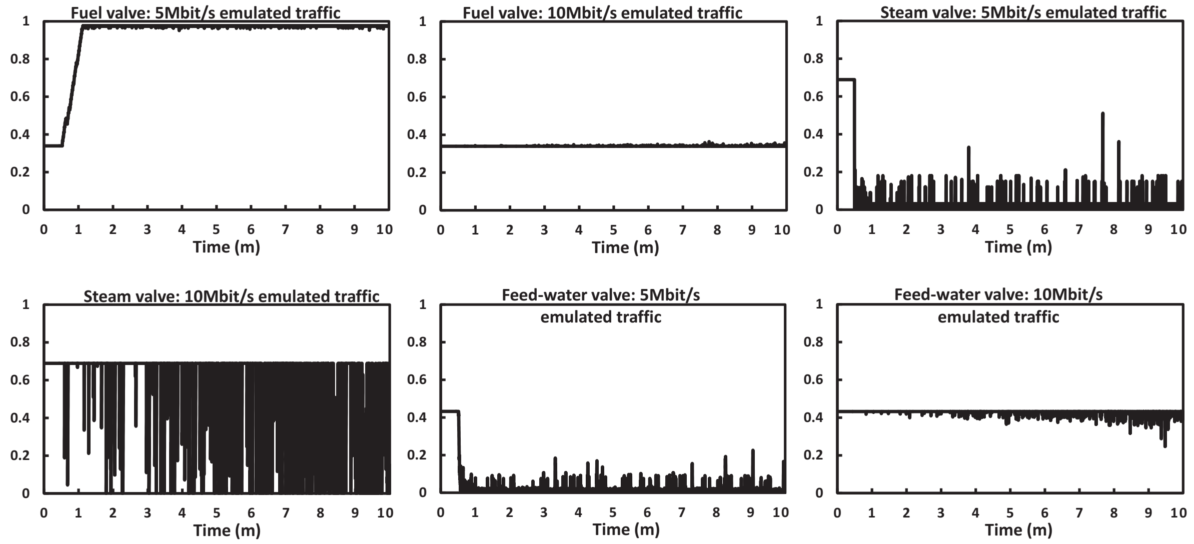

Figure 5: Effect of background traffic on plant valve positions for 100ms TS, 0s emulated delay and 1% packet loss

messages sent every second for each valve generate a traffic of around 140Kbit/s, with a total of 420Kbit/s for all three valves. Nevertheless, when the background traffic reaches the network capacity fewer messages reach the PLCs which are able to maintain the NVP with only small deviations. However, even in this later extreme case the steam valve is affected by the attacker, as its closing speed is 10 times higher than its opening speed, and counteracting one single packet received from the attacker requires 10 executions of the control code.

1ms Task Scheduling

The previous results have shown that if PLCs have a control code TS of 100ms they are not effective in maintaining the NVP. Moreover, the outcome of the attack is affected only by extreme

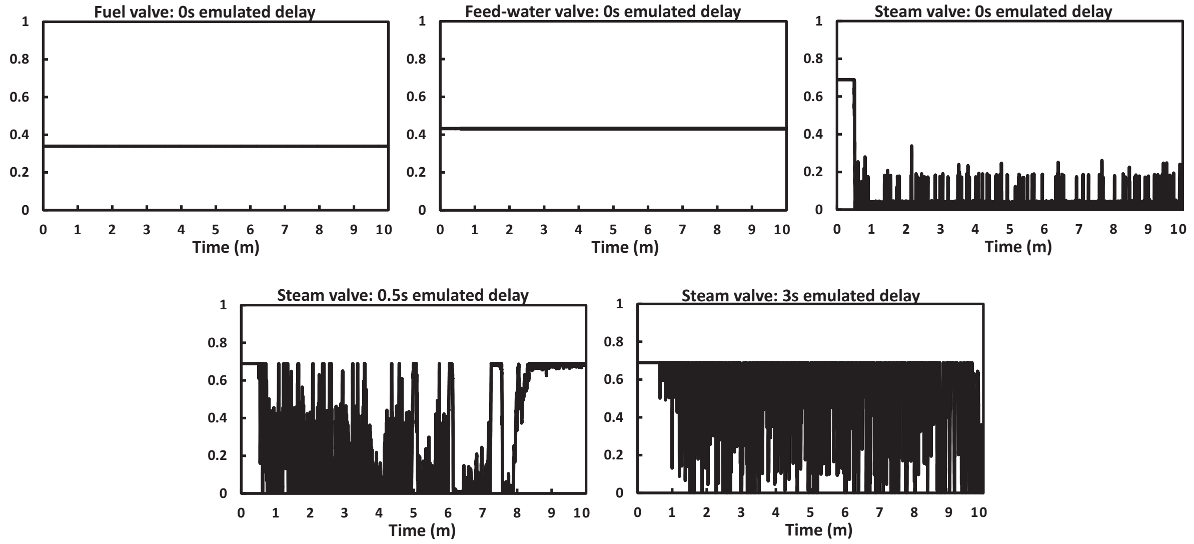

Figure 6: Effect of network delays on plant valve positions for 1ms TS, 1% loss rate and 2.5Mbit/s background traffic

cases of network delays and background traffic. For this reason we have also experimented with a TS of 1ms that could significantly improve the reaction of the PLCs.

For a TS of 1ms, even in the case of 0s emulated delay, the attacker is only able to produce insignificant deviations from NVP for fuel and feed-water valves, as shown in Figure 6. Nevertheless, the steam valve is still affected by the attack and the attacker is able to bring it to the AVP. However, in this case larger network delays show a significant effect on the steam valve, as the attacker is unable to maintain the steam valve in the AVP for delays larger than 3s.

As the fuel and feed-water valves present only insignificant changes already shown in Figure 6, we further focus our attention on the steam valve. We have repeated the experiments for different packet losses and background traffic, with the results shown in Figure 7. As in the case of the previous results, larger packet losses do not improve the response of the PLCs. Nevertheless, the extreme case of 10Mbit/s for background traffic ensures small deviations from the NVP for the first 5 minutes of the experiment. In the second half more packets reach the PLC that produce larger and larger deviations and finally after 8 minutes are able to bring the valve into the AVP.

The results from this sub-section have shown that the attacker is able to affect the NVP for all three valves if the PLCs use a TS of 100ms. Within this setting network delays and background traffic are major factors that influence the attack, as also illustrated in Table 1. Nevertheless, the attacker is still able to produce major deviations from NVP for valves with an opening and closing speed difference. For this reason, a better solution would be to provide a smaller TS combined with an equal opening and closing speed of valves. As shown by the results, with a smaller TS, the fuel and feed-water valves experience insignificant deviations from NVP with minimum emulated delays and background traffic. In contrast, the speed difference of the steam valve still causes major deviations from NVP, that are reduced by only extreme network delays of 3s and a background traffic of 10Mbit/s. Lowering the value of the TS below 1ms could be considered a measure for a more resilient control code. However, this is only possible if the control code's execution time is smaller than this value.

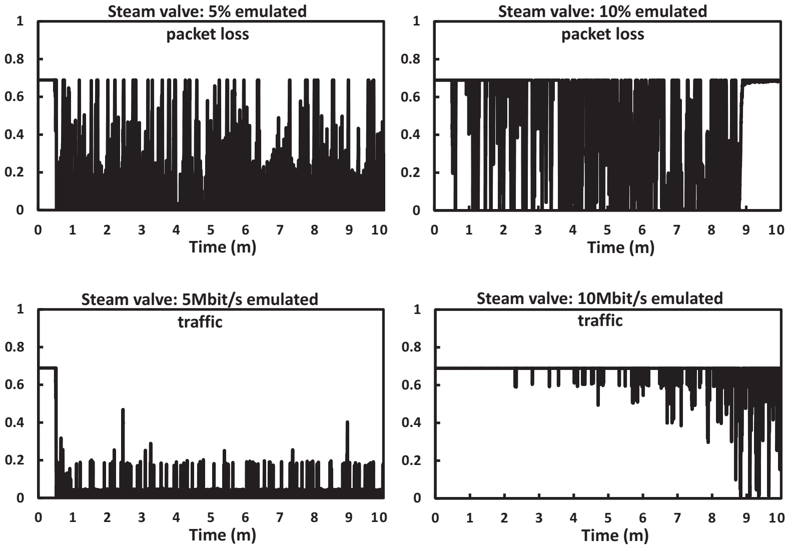

Figure 7: Effect of packet loss and background traffic on the steam valve position for 1ms TS and 0s emulated delay

5.2 The Effect of the Cyber Attack on the Steam Pressure

In the previous sub-section we have shown that the attacker can cause major deviations from the NVP for all three valves. In this sub-section we analyze the effects that the previously presented deviations have on the steam pressure. Following the same experimental strategy, we have first recorded the steam pressure for a TS of 100ms followed by a TS of 1ms.

100ms Task Scheduling

The results from Figure 8 show that for an emulated network delay of 0s, the attacker is able to increase the steam pressure above 250 kg/cm^2 after only 2.5 minutes the attack is started. Moreover, the attacker is able to bring the process into a critical state even for a network delay of 0.5s. Only extreme network delays of 3s show a major impact and prevent the successful outcome of the attack, although the attacker is still able to produce major deviations from NVP, as previously shown in Figure 3. With larger packet losses the process reaches the critical state after 4 minutes for a 5% loss and 6 minutes for 10% loss, as shown in the same figure. The reason for this behavior is that all three valves still experience major deviations from their NVP, as shown in Figure 4, which causes the pressure to increase immediately after the attack is started. The background traffic affects the outcome of the attack only in extreme cases when it reaches the maximum network capacity. Otherwise, the attacker is able to open and close the valves, as shown in Figure 5, and to bring the process into the critical state.

1ms Task Scheduling

By decreasing the TS time to 1ms the attacker is not able to reach his goal in neither of the settings included in this study (Figure 9). Nevertheless, it is still able to increase the steam pressure to a maximum value of 234 kg/cm^2 for minimal network delay, packet loss and background traffic. The reason behind this is that the attacker is still able to completely close the steam valve, although it is able to cause only negligible deviations from the NVP for the other two valves, as shown in Figure 6.

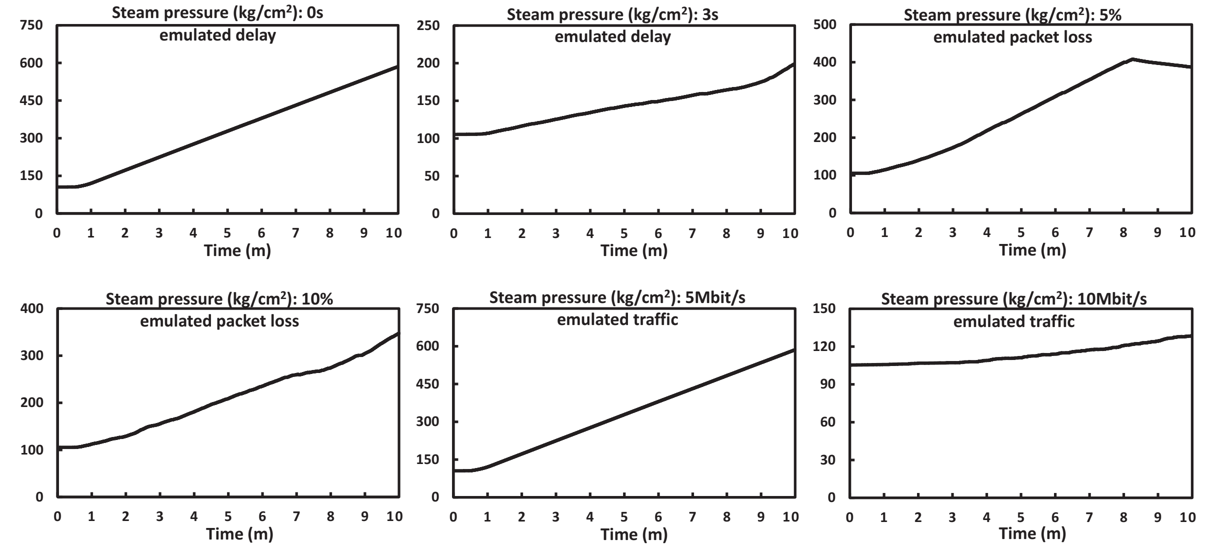

Figure 8: Effect of network delays, packet loss and background traffic on the steam pressure for 100ms TS

Table 1: Average valve positions and maximum pressure (P) during cyber attacks

		Fuel valve		Steam valve		Feed-water valve		Max P (kg/cm^2)	
		(Target: 0.34)		(Target: 0.68)		(Target: 0.43)		(Target: 108)	
Parameters		100ms	1ms	100ms	1ms	100ms	1ms	100ms	1ms
Delay (s)	0	0.93	0.34	0.04	0.05	0.03	0.433	585	234
	0.5	0.75	0.34	0.11	0.31	0.21	0.433	459	173
	3	0.4	0.34	0.25	0.51	0.39	0.433	198	131
Loss (%)	5	0.84	0.34	0.1	0.3	0.14	0.433	407	204
	10	0.61	0.34	0.17	0.46	0.2	0.433	347	145
Traffic (Mbit/s)	5	0.92	0.34	0.04	0.05	0.03	0.433	586	234
	10	0.34	0.34	0.55	0.67	0.43	0.433	128	110

By increasing the emulated network delays the maximum pressure induced by the attacker reduces gradually from 173 kg/cm^2 for 0.5s to 131 kg/cm^2 for 3s (Table 1). However, larger packet losses and background traffic do not produce major differences in the outcome of the attack, unless extreme values are used. Nevertheless, a 5% packet loss is still able to reduce the maximum pressure to 204 kg/cm^2, while a 10% packet loss reduces the maximum pressure to 145 kg/cm^2. In the extreme case of 10Mbit/s background traffic the maximum pressure is reduced to 110 kg/cm^2.

The results from this section have shown that by decreasing the TS to 1ms the physical process is able to react more efficiently to cyber attacks. Network delays, packet losses and background traffic have also shown to have an influence on the attack. Nevertheless, by using a TS of 100ms these are able to affect the outcome of the attack only in extreme cases. Consequently, designers should consider using a lower TS whenever possible in order to prevent the successful outcome of similar attacks.

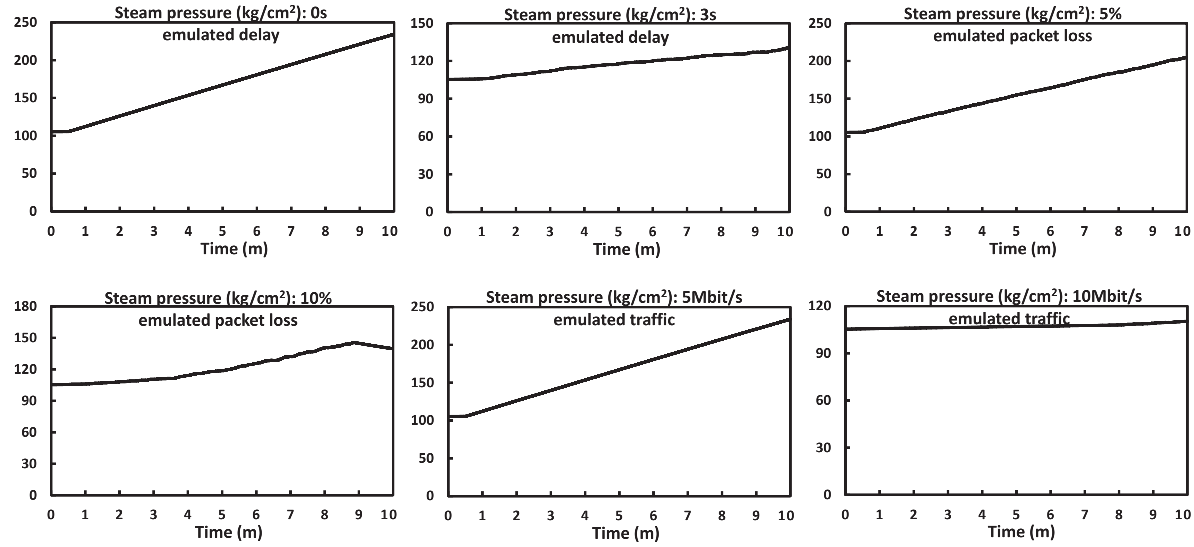

Figure 9: Effect of network delays, packet loss and background traffic on the steam pressure for 1ms TS

6 Conclusion

The study presented in this paper showed that cyber attacks exploiting knowledge on NICS can use regular Modbus packets to bring the physical process into a critical state. Within this scenario we evaluated the impact of network and installation-specific parameters on cyber attacks targeting a power plant. The experimental results showed that while communications parameters such as network delays, packet losses and background traffic have a limited effect on the attack, task scheduling and properties of physical processes, i.e. the speed of control valves, can become effective measures for increasing the resilience of physical processes. The main contribution of this paper is that it identifies two key parameters that could be adopted at design-time to increase the resilience of physical processes confronted with cyber attacks. The first one, i.e. control code task scheduling, provides engineers an efficient mechanism to counterbalance disturbances caused by malicious command packets, while the second one, i.e. the speed of control valves, provides insight into the way that an attacker might manipulate knowledge on physical properties to bring the process into a critical state. Such properties should be taken into account at process design time, which will lead to a more resilient physical process.

Bibliography

[1] S. East, J. Butts, M. Papa, S. Shenoi, A Taxonomy of Attacks on the DNP3 Protocol, in *Proceedings of IFIP Advances in Information and Communication Technology*, 311:67–81, 2009.

[2] T.C. Aseri, N. Singla, Enhanced Security Protocol in Wireless Sensor Networks, *International Journal of Computers Communications & Control*, 6(2):214–221, 2011.

[3] The Symantec Stuxnet Dossier, 2010, http://www.wired.com /images_blogs/threatlevel/2010/11/w32_stuxnet_dossier.pdf

[4] A.S. Brown, SCADA vs. the Hackers - Can Freebie Software and a Can of Pringles Bring Down the U.S. Power Grid?, *Mechanical Engineering*, 124(12), 2002.

[5] I. Nai Fovino, M. Masera, L. Guidi, G. Carpi, An Experimental Platform for Assessing SCADA Vulnerabilities and Countermeasures in Power Plants, in *Proceedings of Human System Interactions*, pp. 679–686, 2010.

[6] I. Nai Fovino, A. Carcano, T. De Lacheze Murel, M. Masera, A. Trombetta, Distributed Critical State Detection System for Industrial Protocols, in *Proceedings of IFIP International Conference on Critical Infrastructure Protection*, pp. 95–110, 2010.

[7] B. Genge, C. Siaterlis, I. Nai Fovino, M. Masera, A Cyber-Physical Experimentation Environment for the Security Analysis of Networked Industrial Control Systems, *Computers & Electrical Engineering*, In Press, 2012.

[8] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, A. Joglekar, An Integrated Experimental Environment for Distributed Systems and Networks, in *Proceedings of the 5th symposium on Operating systems design and implementation*, pp. 255–270, 2002.

[9] C. Siaterlis, A. Garcia, B. Genge, On the Use of Emulab Testbeds for Scientifically Rigorous Experiments, *IEEE Communications Surveys & Tutorials*, PP(99):1–14, 2012.

[10] R.D. Bell, K.J. Åström, Dynamic Models for Boiler-Turbine Alternator Units: Data Logs and Parameter Estimation for a 160MW Unit, *Lundt Institute of Technology, Report TFRT–3192*, Sweden, 1987.

[11] L. Rizzo, Dummynet: A Simple Approach to the Evaluation of Network Protocols, *ACM Computer Communication Review*, 27(1):31–41, 1997.

[12] M Carbone, L. Rizzo, Dummynet Revisited, *ACM SIGCOMM Computer Communication Review*, 40(2):12–20, 2010.

[13] NLANR/DAST, Iperf: The TCP/UDP Bandwidth Measurement Tool, http://sourceforge.net/projects/iperf/

[14] W. Tan, H.J. Marquez, T. Chen, J. Liu, Analysis and Control of a Nonlinear Boiler-Turbine Unit, *Journal of Process Control*, Elsevier, 15(8):883–891, 2005.

[15] C. Queiroz, A. Mahmood, J. Hu, Z. Tari, X. Yu, Building a SCADA Security Testbed, in *Proceedings of the International Conference on Network and System Security*, pp. 357–364, 2009.

[16] C.M. Davis, J.E. Tate, H. Okhravi, C. Grier, T.J. Overbye, D. Nicol, SCADA Cyber Security Testbed Development, in *Proceedings of the North American Power Symposium*, pp. 483–488, 2006.

[17] 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*, April, 2010.

[18] A. Cárdenas, S. Amin, Z.S. Lin, Y.L. Huang, Chi-Y. Huang, S. Sastry, Attacks Against Process Control Systems: Risk Assessment, Detection, and Response, in *Proceedings of the ACM Symposium on Information, Computer and Communications Security*, pp. 355–366, 2011.