Investigating the Effect of Network Parameters on Coordinated Cyber Attacks Against a Simulated Power Plant

Béla Genge and Christos Siaterlis

Joint Research Centre, European Commission
Institute for the Protection and Security of the Citizen
Via E. Fermi, 2749, Ispra (VA), 21027, Italy
{bela.genge, christos.siaterlis}@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 these systems, today we still lack the proper understanding of the effects that cyber attacks have on NICS. In this paper we use our previously developed framework to study the effects of network parameters, i.e. delay, packet losses and background traffic, on coordinated cyber attacks against NICS. Coordinated attacks rely on several infected hosts to disrupt the normal functionality of the system. Within the context of NICS we consider multiple infected control hardware, a highly similar setting to the recently reported Stuxnet worm, the first malware specifically designed to attack NICS. Furthermore, we assume that the coordinator is located outside the system, in the Internet, from where it launches attacks by sending packets to each infected control hardware. The main goal of the attacker is to bring the physical process into a critical state, i.e. dangerous, or more generally unwanted state of the system. For the physical process we used the Boiling Water Power Plant (BWPP) model developed by Bell and Åström.

Keywords: Coordinated attack, Networked Industrial Control Systems, network parameters, Boiling Water Power Plant

1 Introduction

Modern Critical Infrastructures (CIs), 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 reduction 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 accurately 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]. For example, the recently reported

Stuxnet worm [3] is the first malware that is 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; it has served as a wakeup call for the international security community.

As already highlighted by previous research [4], coordinated attacks have a much greater impact on the target system than non-coordinated ones. In a coordinated setting the attacker relies on several infected hosts to disrupt the normal functionality of the system. The recently reported attack on Twitter [5], where a hacker used thousands of infected hosts to launch a DoS attack, has demonstrated just how powerful these attacks can be. Consequently, in this paper we use our previously developed framework [6] to study the effects of network parameters, i.e. delays, packet losses, background traffic, on coordinated cyber attacks against NICS. The coordinator, located outside the power plant, in the Internet, uses multiple infected control hardware to bring the system into a *critical state*, i.e. dangerous, or more generally unwanted state of the system [7]. The control hardware is infected with malicious code and is able to receive commands from the coordinator, which is a reasonable assumption if we consider that the Stuxnet malware showed a similar behavior. The attack scenario was implemented with our previously developed framework [6] that uses simulation for the physical components and an emulation testbed based on Emulab [8] in order to recreate the cyber part of NICS, e.g., SCADA servers, corporate network, etc.

The paper is structured as follows. Our study is presented in the context of other related approaches in Section 2, followed by a short overview of our previous work in Section 3. The experimental scenario and setup are presented in Section 4, followed by the analysis of coordinated attacks involving a Boiling Water Power Plant in Section 5. Conclusions are presented 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.* [10]. 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.* [11] 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.* [12] proved that a DDoS attack against communication nodes between controllers and sensors causes the PLCs to take wrong decisions based on old sensor values. Finally, we mention the work of Cárdenas, *et al.* [14] that didn't only document the effect of DoS attacks on sensors, but also proposed a new detection mechanism together with possible countermeasures.

The previously mentioned approaches 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.* [13] that proposed an experimental platform for studying the effects of cyber attacks against NICS. In their paper the authors described several attack scenarios, including DoS attacks and worm infections that send Modbus packets to control hardware. Although the authors provided a wide range of countermeasures, they did not identify communication parameters that affect the outcome of the attacks.

3 Experimentation Framework Overview

The experimentation framework developed in our previous work [6] follows a hybrid approach, where the Emulab-based testbed recreates the control and process network of NICS, including Programmable Logical Controllers (PLCs) and SCADA servers, and a software simulation reproduces the physical processes. The architecture, as shown in Fig. 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 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

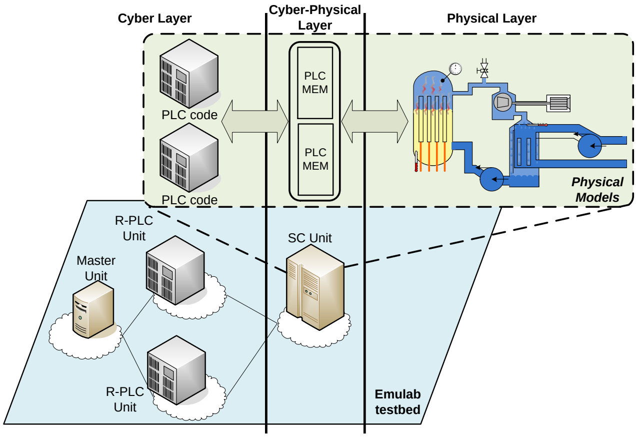


Fig. 1: Experimentation framework architectural overview

execution time is strongly coupled to the timing service of the underlying operating system (OS). As the OS uses multitasking, achieving hard real-time is difficult without the use of kernel drivers. However, soft real-time is achieved by allowing a certain deviation from the OS clock. The cyber layer is recreated by an emulation testbed that uses the Emulab architecture and software [8] 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 implemented 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 PLC). The cyber-physical layer incorporates the PLC memory, seen as a set of registers typical to PLCs, and the communication interfaces that glue together the other two layers. Memory registers provide the link to the inputs (e.g. valve position) and outputs (e.g. sensor values) of the physical model.

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 Description of the Experimental Setup

4.1 Scenario

As pointed out by Cárdenas, *et al.* [14] attacks targeting the minimum/maximum value of parameters/control variables are the ones that can damage the process in relatively short time periods. Such attacks cause the accumulation of products (e.g. steam, water, fuel) by completely opening valves that feed products into process units and completely closing valves that free products from the process units. Our employed adversary model followed the same procedure to damage the physical process.

In the implemented scenario the attacker interacts with PLCs by sending legitimate Modbus packets to close/open specific valves. The attacker is located in the Internet and uses TCP connections to communicate with infected PLCs. Identifying the attack vector that could compromise the system to enable such a scenario is not the main focus of this study. However, we should also mention that the Stuxnet worm together with other studies such as the one performed by Nai Fovino, *et al.* [13], showed that such scenarios are possible in real settings. For instance, the attack reported in 2010 on Google's stations [15] is a clear example of how malicious software is able to exploit a Web browsers vulnerability in order to infect the entire corporate network of a large organization. Similarly for an industrial installation, once the malware is installed within the corporate network, it could spread to the process and control networks and it could compromise network protection mechanisms, e.g. firewalls, in order to give access to an adversary, i.e. *coordinator*, located outside the system.

The main target of the attacker was a power plant, integrated into our framework with the Boiling Water Power Plant (BWPP) model developed by Bell and Åström in [9]. This models a 160MW oil-fired electric power plant based on the Sydsvenska Kraft AB plant in Malmö, Sweden. Within the context of this model the attacker is able to control 3 valves: fuel valve, steam valve, and feed water valve. The desired critical state is given by the value of the pressure inside the steam drum. In other words, the goal of the attacker is to increase the pressure up to a specific value, representing the critical state, which can cause the plant to fail, shut down or even explode. The attacker achieves his goal by infecting PLCs that control the 3 valves and by coordinating the attack with packets sent remotely to each PLC.

4.2 Experimental Setup

The following experiments were 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. As shown in Fig. 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 Master unit and 1 host to run the malicious coordinator software. Within the Emulab testbed we emulated communication delays, packet losses and background traffic in order to recreate a dynamic and unpredictable environment such as the Internet. For emulating communication delays and packet losses we used *Dummynet* and for the background traffic we used UDP packets generated with *iperf*. Dummynet and iperf are running on the malicious coordinator and the infected Master unit, as shown in Fig. 2. Additionally, we used two 10Mb/s networks to emulate the limited bandwidth in the Internet (*Lan2*) and the communication limitations of PLCs (*Lan1*). The communication between R-PLCs and the SC unit was implemented with a 100Mb/s Lan (*Lan0*) to provide maximal performances for the interaction between R-PLC units and the BWPP model.

The main role of the SC unit was to run the BWPP model and to enable its interaction with the other components. Within the previously described scenario, each R-PLC unit controls a specific valve. Thus, R-PLC unit 1 controls the fuel valve, R-PLC unit 2 controls the steam valve and, finally, R-PLC unit 3 controls the feed water valve. The attack initiation commands are transmitted by the malicious coordinator using TCP connections and are forwarded by the infected Master unit as Modbus packets. This way, we emulate the functionality of other infected units in the process network that collaborate with the coordinator to succeed in the execution of the attack.

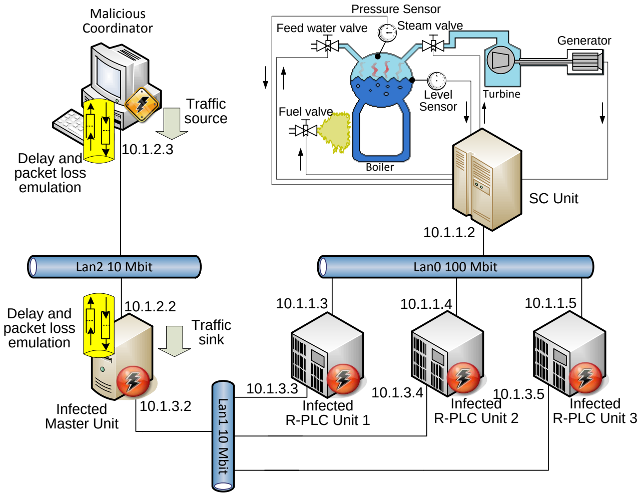


Fig. 2: Experimental setup

5 Attacks and Analysis

In this section we show that communication delays, packet losses and background traffic have a major impact on the success of coordination-based attacks. For this purpose we consider two settings. In the first setting the goal of the attacker is to bring the BWPP into a critical state where the value of the pressure is 249.7285 kg/cm^2. This is more than twice the value of a typical operating point (i.e. 105.006 kg/cm^2) and was obtained by running the model for a total of 260s with the fuel valve completely opened, the steam valve completely closed and the feed water valve set to 0.433. Consequently, in order to bring the BWPP into this state, the attacker needs to open the fuel valve exactly 79s before closing the steam valve, while running with the feed water valve set to 0.433 at all times. The attacker also calculates that after receiving the initiate commands, PLCs need to run the malicious code for 3 minutes in order to bring the BWPP into the critical state. For the second setting we consider that lower *precisions*, i.e. deviations from a fixed steam pressure value, can also bring the plant into a critical state. We show that this consideration increases the attacker's success rate, however, for extreme settings of communication delay, packet losses and background traffic, the coordinated attack still represents a challenge to the attacker.

The parameters we consider for the following experiments are communication delays, packet losses and background traffic. For communication delays we used the following values: 0s, 0.1s, 0.5s, 1s, 3s, 6s and 9s. For packet losses we used two rates: 1% and 5%. Finally, for background traffic we used: 2.5Mb/s, 5Mb/s, 7.5Mb/s and 10Mb/s. For each configuration setting, representing a combination of communication delay, packet loss rate and background traffic we ran 20 experiments, with a total of 1120 experiments executed in 112 hours.

5.1 Effect of Communication Delays and Packet Losses

Communication delays and packet losses between the coordinator and the compromised Master unit were emulated with the *Dummynet* software. We emulated 7 different delays up to 9s and two different packet loss rates: 1% and 5%. As previously mentioned, we assumed that the critical state includes a steam pressure of 249.7285 kg/cm^2.

Within this context we measured a maximal success rate of 70% and a minimal success rate of 0%. The results show that even for zero communication delays the average success rate does not exceed 51.25%. More specifically, this means that from 20 attempts, an average of 10 attempts will fail to bring the BWPP into the desired state. What is even more surprising is that for the majority of cases we measured a higher success rate for a larger loss rate. An explanation for this behavior is the reduced number of packets that are sent by the coordinator as opposed to the number of packets generated for the background traffic. As the number of packets transmitted between stations also affect the delay between packets transmitted by the coordinator, a higher packet loss rate translates to a more reduced number of packets and effectively to smaller delays between coordinator packets. However, this statement is only valid for delays smaller than 1s. For larger delays the success rate drops to 0% as the critical state can not be reached even after PLCs receive the initiate commands. These results are depicted in Fig. 3.

	Attack success rate	
Delay	**1% packet loss rate**	**5% packet loss rate**
0s	45%	51.25%
0.1s	20%	26.25%
0.5s	40%	40%
1s	45%	31.25%
3s	0%	2.5%
6s	0%	0%
9s	0%	0%

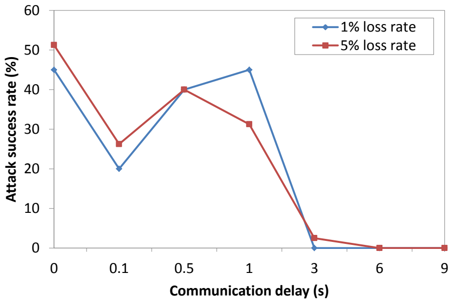
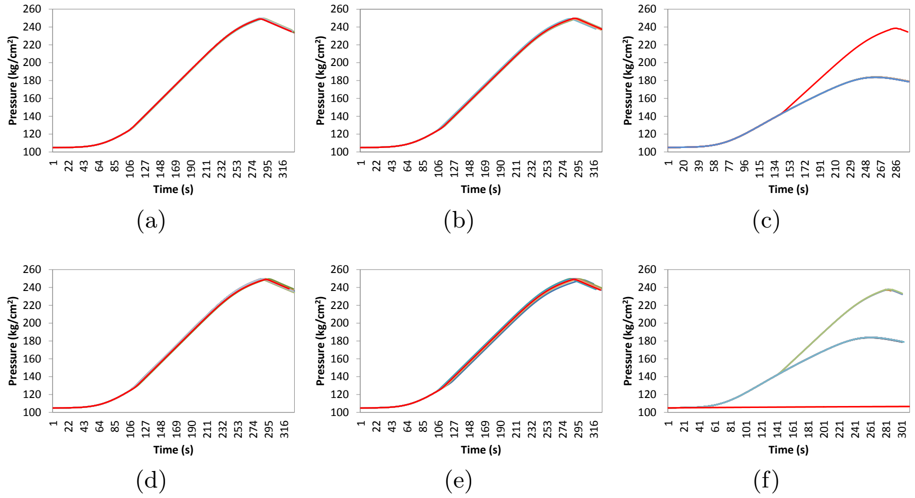


Fig. 3: Effect of communication delays and packet losses on the attack success rate (average background traffic)

For a better understanding of the behavior of the physical process, in the following we provide several figures illustrating the steam pressure for 6 different settings. The behavior of the process for communication delays of 0s, 1s and 9s and packet loss rates of 1% and 5% is shown in Fig. 4. As shown in Fig. 4 (b), a delay of 1s introduces only small variations that are barely visible. On the other hand, larger delays such as 9s illustrated in Fig. 4 (c), lead to connection time-outs that in turn cause a successful execution of commands in only one experiment (out of 20), that was illustrated with a red line. In order to illustrate the effect of a 5% loss rate we have also included Fig. 4 (d), (e) and (f). For this setting variations are more visible. Nevertheless, in case of Fig. 4 (d) and (e) more than 50% of the attacks are successful. For a 9s delay (Fig. 4 (f)) we notice a setting marked with a red line in which none of the PLCs receive the commands to initiate the attack.



(a) (b) (c)

(d) (e) (f)

Fig. 4: Effect of delays and packet losses on the steam pressure for a constant background traffic of 2.5Mb/s: (a) 1% packet loss and 0s delay; (b) 1% packet loss and 1s delay; (c) 1% packet loss and 9s delay; (d) 5% packet loss and 0s delay; (e) 5% packet loss and 1s delay; (f) 5% packet loss and 9s delay

By analyzing the previous results we realize that achieving a 100% success rate for a fixed pressure in a limited time interval is a difficult task. As the attack scenario is highly time critical, emulated network delays and packet losses introduce additional delays to the already existing ones caused by communications and OS task switches. Based on these facts we can clearly state that a coordinated attack launched from outside a power plant has a low success rate (an average of 51.25% for 0s emulated delay) in case of time-critical scenarios.

5.2 Effect of Communication Delays and Background Traffic

The *iperf* software was used to generate UDP background traffic with four different configurations: 2.5Mb/s, 5Mb/s, 7.5Mb/s and 10Mb/s, that was the maximum capacity of *Lan2* (see Fig. 2). This way we were able to simulate real Internet conditions with a permanent background traffic that could also introduce additional delays and thus interfere with the outcome of the attack.

In order to analyze the effect of background traffic on the attack success rate we assumed that the critical state includes the previously discussed steam pressure of 249.7285 kg/cm^2. Additionally, for every configuration we considered an averaged packet loss rate with the results shown in Fig. 5. As expected, the background traffic also influences the attacker's success rate. We clearly see that the highest success rate is achieved for 2.5Mb/s followed by 5Mb/s. A background traffic of 10Mb/s introduces larger delays in *Lan2* (with a 10Mb/s capacity) and reduces the success rate to maximum 20%.

	Attack success rate			
Delay	**2.5Mb/s traffic**	**5Mb/s traffic**	**7.5Mb/s traffic**	**10Mb/s traffic**
0s	52.5%	67.5%	55%	17.5%
0.1s	32.5%	25%	15%	20%
0.5s	52.5%	45%	47.5%	15%
1s	52.5%	52.5%	27.5%	20%
3s	0%	0%	0%	5%
6s	0%	0%	0%	0%
9s	0%	0%	0%	0%

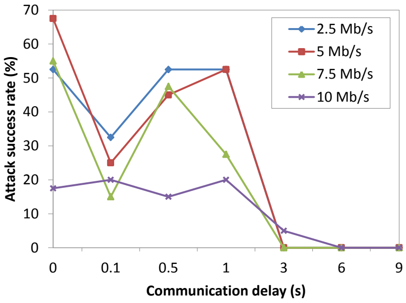
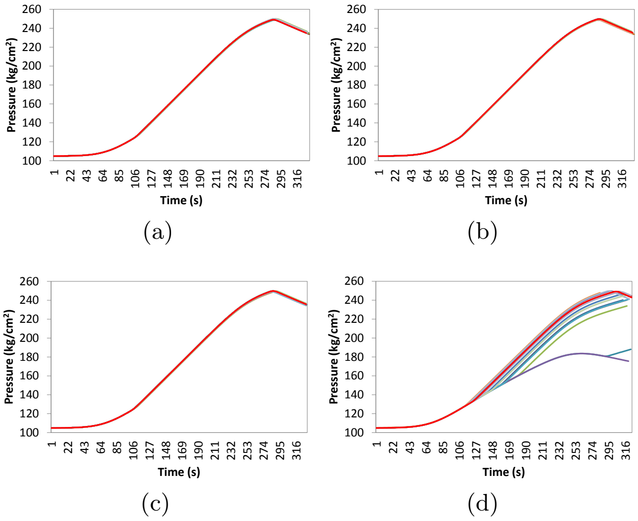


Fig. 5: Effect of communication delays and background traffic on the attack success rate (average packet loss rate)

The behavior of the plant in terms of steam pressure for each background traffic configuration is shown in Fig. 6 (a), (b), (c) and (d). In order to illustrate the effect of background traffic we considered a communication delay of 0s and a 1% packet loss rate in all four settings. These figures clearly show that a background traffic lower than 10Mb/s does not have a major impact on the behavior of the plant. The explanation for this is the low number of commands the attacker needs to send to the remote PLCs in order to initiate the attack. Furthermore, if we compare the effect of background traffic (Fig. 6) with the effect of packet losses (Fig. 4) we realize that packet losses have a greater impact than background traffic. Nevertheless, by increasing the background traffic to 10Mb/s the impact becomes immediately visible as the additional delays affect the timing of the commands received by each PLC. Based on these results we can clearly state that the impact of background traffic on the attack success rate is mainly minor. However, for a background traffic that is close to the network capacity the success rate drops to 20% or even 5% for delays larger than 3s.



(a) (b)

(c) (d)

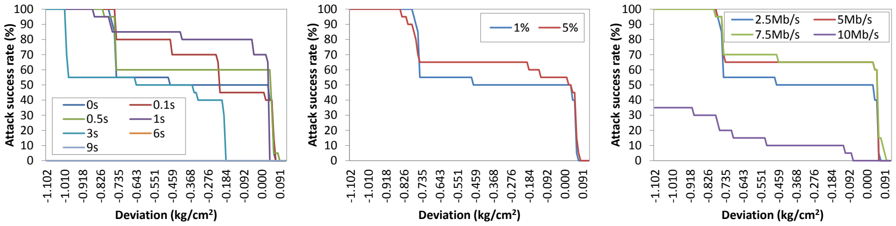
Fig. 6: Effect of delays and background traffic on the steam pressure for a constant delay of 0s and a constant packet loss rate of 1%: (a) 2.5Mb/s background traffic; (b) 5Mb/s background traffic; (c) 7.5Mb/s background traffic; (d) 10Mb/s background traffic

5.3 Effect of Lower Attack Precisions

In the previous sub-sections we assumed that the target steam pressure of the attack is fixed to 249.7285 kg/cm^2. For this setting we measured an average success rate of 51.25% (for 0s emulated delay), with a minimal success rate of 0% and a maximal success rate of 70%. The previous results have also shown that reaching a fixed critical state is a rather difficult task for a coordinator located outside the power plant. However, if the target steam pressure does not require such a high *precision*, i.e. deviation from a fixed value, then the attacker's success rate could suffer major changes. Fig. 7 illustrates these changes in terms of communication delays (Fig. 7 (a)), packet losses (Fig. 7 (b)) and background traffic (Fig. 7 (c)).

As shown in Fig. 7 (a), a critical state with a lower precision increases the attacker's success rate up to 100%, for a precision of 1 kg/cm^2. Nevertheless, larger delays (3s) still have a negative effect on attacks. More specifically, a delay of 3s decreases the success rate to 55%, while delays of 6s and 9s decrease the success rate to 0%. The effect of packet losses are also negligible if we consider lower precisions, as shown in Fig. 7 (b). In this case also the success rate increases up to 100% for a precision of 1 kg/cm^2. The same figure also shows that for the majority of cases a 5% loss rate leads to a higher success rate. As already mentioned in the previous sub-sections the reason behind this behavior is the low number of packets that the attacker uses to initiate the attack, as opposed to the high number of packets available for the background traffic. Finally, as shown in Fig. 7 (c), the effect of background traffic seems to be the most persistent even for lower precisions, as the success rate remains below 40% for a background traffic of 10Mb/s. Nevertheless, the attacker is able to achieve a 100% success rate for a precision of 1 kg/cm^2 and a lower background traffic.

The results from this sub-section have shown that if the critical state allows a slight deviation from the fixed steam pressure then the success rate of the attacker increases dramatically. However, the success rate is still affected by specific delays, packet losses and background traffic, as all of these parameters directly affect the timing between packets. Furthermore, extreme configurations still manage to decrease the success rate from 100% to below 40%.



(a) (b) (c)

Fig. 7: Effect of various attack precisions on the attack success rate: (a) communication delays; (b) packet losses; (c) background traffic

6 Concluding Remarks

In this paper we have analyzed the effects of network parameters on coordinated attacks against a Boiling Water Power Plant (BWPP). The experimental results prove that a coordinated attack where timing between commands is critical has a low success rate (an average of 51.25% for 0s emulated delay). Furthermore, such attacks are highly sensitive respect to communication delays, packet losses and background traffic. Nevertheless, the attacker's success rate increases significantly if the critical state allows a certain deviation from the target parameters. The experimental results also show that while a small deviation might increase the success rate up to 100%, there are configurations in which even these do not ensure a 100% success rate. Such configurations include communication delays larger than 3s and a high network background traffic, close to the network capacity.

References

1. Nai Fovino, I., Carcano, A., Masera, M., Trombetta, A: An experimental investigation of malware attacks on SCADA systems. International Journal of Critical Infrastructure Protection, vol. 2, no. 4, pp. 139–145 (2009)
2. East, S., Butts, J., Papa, M., Shenoi, S.: A Taxonomy of Attacks on the DNP3 Protocol. IFIP Advances in Information and Communication Technology, vol. 311/2009, pp. 67–81 (2009)
3. The Symantec Stuxnet Dossier (2010), http://www.wired.com/images_blogs/threatlevel/2010/11/w32_stuxnet_dossier.pdf
4. Tan, Y., Sengupta, S., Subbalakshmi, K.P.: Analysis of Coordinated Denial-of-Service Attacks in IEEE 802.22 Networks. IEEE JSAC Special Issue on Cognitive radio Networking and Communications, vol. 29 (4), pp. 890–902 (2011)
5. Botnet Twitter Attack (2009), http://www.usatoday.com/tech/news/2009-08-06-twitter-attack_N.htm
6. Genge, B., Nai Fovino, I., Siaterlis, C., Masera, M.: A Framework for Analyzing Cyber-Physical Attacks on Networked Industrial Control Systems. Fifth Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection, To Appear (2011)
7. Nai Fovino, I., Carcano, A., De Lacheze Murel, T., Masera, M., Trombetta, A.: Distributed Critical State Detection System for Industrial Protocol. In Proceeding of the Forth Annual IFIP International Conference on Critical Infrastructure Protection, pp. 95–110. Washington DC, USA (2010)
8. White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M. Hibler, M., Barb, C., Joglekar, A.: An integrated experimental environment for distributed systems and networks. In Proc. of the Fifth Symposium on Operating Systems Design and Implementation, pp. 255–270 (2002)
9. Bell, R. D., Åström, K. J.: 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)
10. Queiroz, C., Mahmood, A., Hu, J., Tari, Z., Yu, X.: Building a SCADA Security Testbed. In Proc. 3th NSS, pp. 357–364 (2009)
11. Davis, C.M., Tate, J.E., Okhravi, H., Grier, C., Overbye, T.J., Nicol, D.: SCADA Cyber Security Testbed Development. In Proc. NAPS, pp. 483–488 (2006)
12. Chabukswar, R., Sinopoli, B., Karsai, G., Giani, A., Neema, H., Davis, A.: Simulation of Network Attacks on SCADA Systems. First WSCS, April (2010)
13. Nai Fovino, I., Masera, M., Guidi, L., Carpi, G.: An Experimental Platform for Assessing SCADA Vulnerabilities and Countermeasures in Power Plants. In Proc. HSI, pp. 679–686 (2010)
14. Cárdenas, A., Amin, S., Lin, Z.S., Huang, Y.L., Huang, C.Y., Sastry, S.: Attacks Against Process Control Systems: Risk Assessment, Detection, and Response. In Proc. ASIACCS, pp. 355–366 (2011)
15. Google Aurora attack (2010), http://www.wired.com/threatlevel/2010/01/operation-aurora/