AMICI: An Assessment Platform for Multi-Domain Security Experimentation on Critical Infrastructures

Béla Genge, Christos Siaterlis, and Marc Hohenadel

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, marc.hohenadel}@jrc.ec.europa.eu

Abstract. This paper presents AMICI, a new Assessment/analysis platform for Multiple Interdependent Critical Infrastructures (CIs). Its architecture builds on our previous work and uses Emulab to recreate ICT software and hardware components and Simulink to run the physical process models. Our previous framework is extended with software components to provide a set of capabilities that would enable the analysis of complex interdependencies between multiple CIs: flexible integration of multiple physical process models; opened architecture to enable interaction with ad-hoc software; support experimentation with real software/-malware; automated experiment management capabilities. The applicability of the approach is proven through a case study involving three CIs: ICT, power grid and railway.

Keywords: Critical Infrastructure, security, experimentation, testbed

1 Introduction

As shown by recent studies [1], today's Critical Infrastructures (CIs) are highly dependent of each other. In fact, in many cases relationships are bidirectional and the successful operation of one CI might depend on an entire chain of interdependent CIs. On top of that, modern CIs, e.g. power plants, water plants and smart grids, rely on Information and Communications Technologies (ICT) for their operation since ICT can lead to cost reduction, 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 are exposed to significant cyber-threats, as shown by recent events such as Stuxnet [2] and Flame [3].

The complexity and the need to understand these interdependent systems lead to the development of a wide range of approaches for analyzing interdependencies between CIs [4–6]. Although these can effectively model and analyze bidirectional relationships at a conceptual level, in practice the propagation of disturbances and their magnitude might depend on parameters that are difficult to model. This aspect is especially true in ICT, where it is a well-known fact that models might recreate normal operations, but they fail to capture the complexity of real components, e.g. complex interactions between heterogeneous software/malware and hardware [7].

Existing approaches for cyber security experimentation with CIs either focus on a specific CI [8–10], or they do not enable experimentation with real software/-malware [11, 12], that nowadays is a fundamental requirement for conducting experiments with ICT infrastructure [13]. Based on these facts in this paper we propose a new approach for conducting multi-domain security experiments on CIs. The approach builds on the framework developed in our previous work [14, 15] and extends it with software modules in order to enable experimentation with more than one CI. The final framework, called *AMICI* (*Assessment/analysis platform for Multiple Interdependent Critical Infrastructures*), uses simulation for the physical components and an emulation testbed based on Emulab [16, 17] in order to recreate the cyber part of CIs, e.g. BGP routing protocols, SCADA (Supervisory Control And Data Acquisition) servers, corporate network. The use of simulation for the physical layer is a very reasonable approach due to small costs, the existence of accurate models and the ability to conduct experiments in a safe environment. The argument for using emulation for the cyber components is that the study of the security and resilience of computer networks would require the simulation of all the failure related functions, most of which are unknown in principle. The novelty of the proposed approach is that it brings together a wide range of functionalities, most of which are missing in related approaches [8–12]. These include flexible experimentation with multiple CIs, support of real software and malware, and automated experiment management capabilities. The *flexibility* and *real* functionalities are ensured through the use of real hardware, e.g. PCs, switches, routers, and real Operating Systems that can run generic software/malware together with typical network protocols. Lastly, the *automated* functionality is inherited from Emulab and includes a wide range of sub-functionalities such as experiment configuration, event scheduling, and image management [14, 15]. The approach is validated through a case study showing the interdependencies between three CIs: the power grid, the railway system and the ICT infrastructure.

The rest of the paper is structured as follows. A discussion on the requirements for the design of AMICI, together with the proposed architecture and implementation are detailed in Section 2. The approach is validated in Section 3 through a case study that includes a cyber attack on the ICT infrastructure and a disturbance on the power grid that propagates to the railway system, causing an immediate stop of several trains. The paper concludes in Section 4.

2 Design and Architecture of AMICI

2.1 Design Requirements

Ideally, an experimentation framework for multi-domain security research would support the execution of complex, large scale and disruptive experiments using rigorous scientific methods. The implemented functionalities should not only

Table 1: Required functionalities for multi-domain security experimentation

ID	Functionality
F_1	Support a wide range of physical process models, e.g. power systems, railway
F_2	Support multiple models in parallel and enable data exchange between them
F_3	Support typical ICT components, e.g. SCADA servers, PLCs, Modbus protocols
F_4	Support real software and malware
F_5	Support interaction of models with ad-hoc software
F_6	Support automated and multi-user experiment management

support a wide range of physical processes, e.g. industrial systems, transportation, healthcare, but should also take into account the presence of ICT and specifically of SCADA components commonly used in the monitoring and control of physical processes. Such components include SCADA servers (Masters), PLCs (Programmable Logic Controllers) and typical industrial protocols such as Modbus. Besides these, today's experimentation frameworks should not be closed and should facilitate their extension together with the addition of other custom or even proprietary software. On top of these, an experimentation framework would also need to include capabilities that facilitate the experimentation process and would support concurrent users at the same time. These capabilities are specific to Internet experimentation testbeds and include a wide range of aspects such as control of the experiment's environment, experiment automation, and secure remote access. For a more detailed presentation on the requirements of an Internet security testbed the reader should consult our previous work [18]. A summary of these requirements is also given in Table 1.

2.2 Overview of Our Previous Work

The framework developed in our previous work [14, 15] was specifically designed to enable experimentation with SCADA systems. It includes one simulation unit to run a model of the physical process and software components to emulate real PLCs and SCADA servers. Communications between the simulation and PLC emulator units are implemented through .NET's binary implementation of RPC/TCP, while communications between PLC and SCADA server emulators are implemented through Modbus/TCP.

The framework currently supports the execution of control code, i.e. emulated PLCs, running sequentially and in parallel to the physical process 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. 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. For the physical process simulator we used Matlab Simulink, since it is a general simulation environment for dynamic and embedded systems and covers a wide variety of physical processes, e.g. power plants, gas plants. From

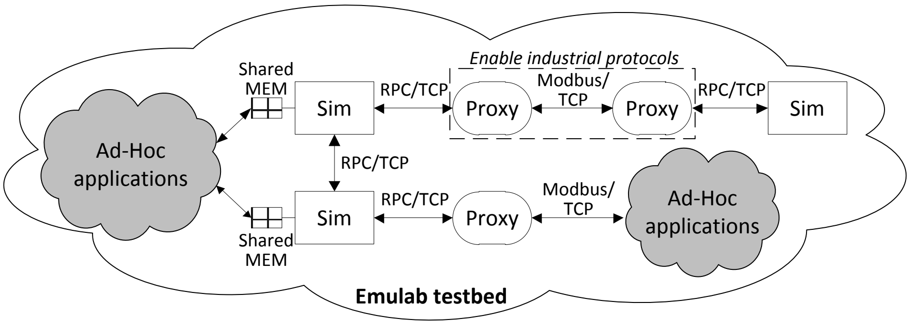


Fig. 1: Architecture of AMICI

Simulink models the corresponding 'C' code is generated using Matlab Real Time Workshop and is integrated into the framework using an XML configuration file.

2.3 Extensions to Our Previous Work and Architecture of AMICI

The architecture of AMICI shown in Fig. 1 is an extension of the framework architecture proposed in our previous work [14, 15]. The main changes made in order to fulfill the previously stated requirements include: (i) addition of an RPC client module in the simulation unit (*Sim*) to enable communications with other *Sim* units; (ii) addition of a shared memory handler module in the *Sim* unit to enable exchange of data between the physical process model and ad-hoc software; and (iii) a new *Proxy* unit that extends our previous PLC emulator with modules allowing it to translate Modbus to RPC and vice-versa. The architecture and its motivation for each unit are detailed in the remaining of this section.

Simulation unit. The main role of the simulation unit (*Sim*) is to run the physical process model in real-time. This is done by coupling the model time to the system time in such a way to minimize the difference between the two. Models are constructed in Matlab Simulink from where the corresponding 'C' code is generated using Matlab Real Time Workshop. These are then integrated using an XML configuration file that is flexible enough so that researchers do not need to modify the code of AMICI. From the *Sim* unit's point of view each model is seen as a set of inputs and outputs. These are mapped to an internal memory region (*I/O MEM*) that is read/written by other software modules as well, e.g. TCCs, RPC. Compared to the previous version, the *Sim* unit allows an open access to its *I/O MEM* by implementing OS level shared memory operations. This way, AMICI enables interaction with ad-hoc software that can write specific model inputs, i.e. OPEN/CLOSE a valve, and can read the status of the model, i.e. measured voltage. Interaction with other *Sim* units is enabled by implementing not only RPC server-side operations but client-side calls as well. By using only the XML configuration file, the *Sim* unit can be configured to read-/write inputs/outputs of models run by remote *Sim* units. These are mapped

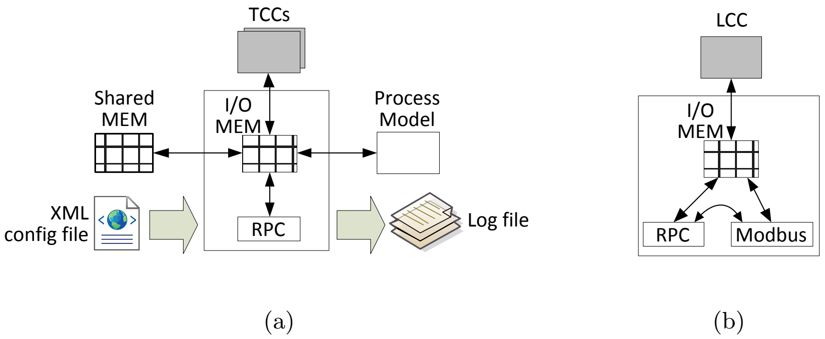


Fig. 2: Detailed architecture: (a) Simulation unit, and (b) Proxy unit

to the inputs/outputs of the model running locally, enabling this way complex interactions between models running in parallel on different hosts.

The *Sim* unit fulfills another important functionality that was previously handled by the SCADA master unit. In AMICI, SCADA server units are implemented as *Sim* units, where the global decision algorithm is the actual physical process model. As the *Sim* unit implements RPC and SCADA servers use industrial protocols, AMICI adopts the *Proxy* unit to map messages from RPC to Modbus and vice-versa. The architecture of the *Sim* unit is given in Fig. 2 (a).

Proxy unit. The *Proxy* unit has several roles within AMICI. At the beginning, its main role was to enable running remote control code through the form of LCCs, enabling this way the integration of more complex PLC emulators. At the same time, it was used to handle Modbus calls coming from SCADA servers and transforming them to RPC calls that were finally sent to the *Sim* unit. AMICI keeps all these capabilities, but it enriches the protocol mapping capabilities of the *Proxy* unit in order to enable running industrial protocols between two *Sim* units. A more detailed architecture of the *Proxy* unit is given in Fig. 2 (b).

2.4 Real-Time Monitoring of Experiments

AMICI uses *Zabbix* [19], an open-source distributed network monitoring and visualization tool, to monitor experiments in real-time. It mainly consists of agents that are installed on the monitored nodes and servers that collect and store data from agents. Zabbix includes built-in monitoring of OS parameters, e.g. CPU, MEM, network traffic, but it also allows defining custom parameters. Such parameters are defined in the `zabbix_agentd.conf` file and have a unique ID that is used by the Zabbix server in the periodical pooling of agents. In AMICI the *Sim* unit writes the model input and output values for each execution step in a log file. From there, Zabbix agents extract specific parameters and send them to the Zabbix server.

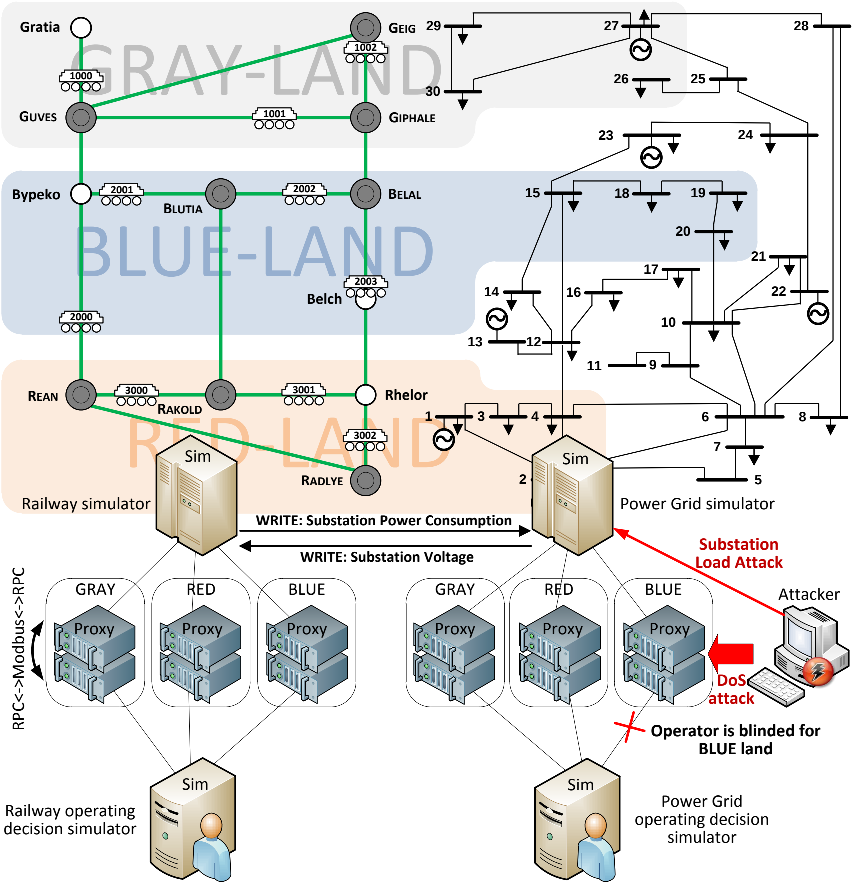


Fig. 3: Experiment setup

3 Case Study

In this section we use the AMICI framework to study the propagation of perturbations between three CIs: the power grid, the railway system and the ICT infrastructure needed to monitor and control them. We show that the power grid and railway system can be highly dependent of each other and in order to ensure the stability of these two, the ICT infrastructure must be intact. We start with a brief presentation of the experiment setup and scenario and then continue with the analysis of the results.

3.1 Description of the Employed Critical Infrastructures

The Power Grid. The power grid employed in this experiment is the well-known IEEE 30-bus model (see Fig. 3 for its graphical representation). It includes

6 generators and 30 substations that deliver power to connected loads through transmission lines. For each substation there is a fixed load and a variable load. Fixed loads are needed to ensure the stability of the grid, while variable loads depend on the power consumed by trains running within the railway system. More specifically, we assume that each railway line, i.e. segment, is connected to one of the grid's substation.

The Railway System. The railway system we employed (see Fig. 3) was constructed from several train models of the type proposed by Ríos and Ramos in [20]. The train model takes into account several realistic aspects of modern transportation systems, e.g. weight, speed, acceleration and deceleration. In their paper, the authors also provide the equations for calculating the instantaneous power consumption of each train. This gives us the possibility to directly connect the output of the model, i.e. power consumption, to the input of the power grid model, i.e. load on each substation. Within this experiment we do not take into account traffic regulation algorithms, as our main focus is illustrating the applicability of AMICI in the study of interdependencies.

The ICT Infrastructure. The ICT infrastructure shown in Fig. 3 is responsible for the monitoring and control of the two infrastructures previously mentioned. For the power grid, the ICT infrastructure includes automated operational algorithms that can detect a change in substation voltage and can issue a command to start/stop backup generators. For the railway system operational algorithms can start/stop specific trains, but in reality there could also be traffic regulation algorithms running on the operator's side.

(Inter)Dependencies. There can be several dependencies between the three CIs previously mentioned, as shown in Fig. 4. First of all, it is clear that the railway system needs to be powered from the power grid. It is also clear that both the railway and the power grid need ICT control to ensure normal operation and that ICT infrastructures need to be powered from the power grid. What is particularly interesting, also depicted in Fig. 4, is that the railway system might have an undesirable effect on the normal operation of the power grid while the later one is subject to a heavy load. In such cases the power grid can be extremely sensitive to additional loads, i.e. starting trains, and if no additional measures are taken by operators, voltages can collapse, leading to other cascading failures. Another aspect highlighted in Fig. 4 is the ICT infrastructure that was split in two: the *Railway ICT* and the *Grid ICT*. Although separated, in practice physical links can be shared between the two, there can be other dependencies that were not taken into account in this experiment.

3.2 Experiment Scenario

For the implemented scenario we defined three hypothetical regions that are common to the power grid and railway CIs (as shown in Fig. 3). These were named *GRAY-LAND*, *BLUE-LAND* and *RED-LAND*. Each substation included in each region powers one specific segment within the railway system. This means that in case voltages drop below an operating threshold, i.e. 0.95 p.u., trains will stop and operators will need to manually restart them. For each region we defined

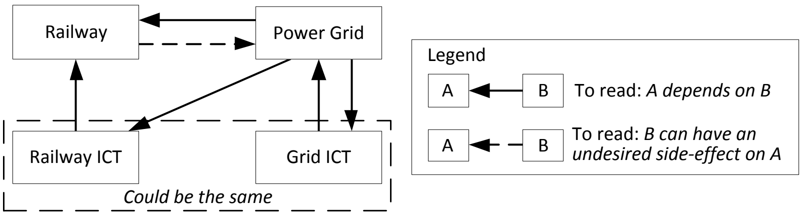

Railway
Power Grid
Railway ICT
Grid ICT
Could be the same
Legend
A
B
To read: *A depends on B*
A
B
To read: *B can have an undesired side-effect on A*

Fig. 4: Possible dependencies between three Critical Infrastructures

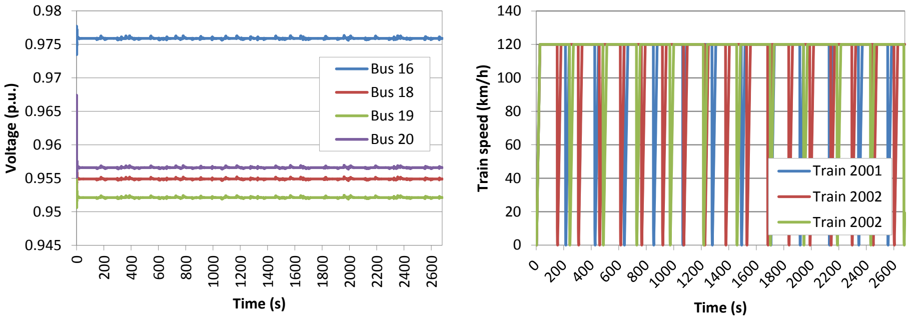
a set of ICT devices and one global operator for each of the two CIs, i.e. power grid and railway system.

The scenario involves an attacker that tries to stop trains running within the *BLUE-LAND* by issuing an attack in two phases. In the first phase the attacker runs a Denial of Service (DoS) against monitoring devices within the *BLUE-LAND* region, in order to inhibit any further data exchange between operators and the physical process. This completely blinds the operators that fail to receive any updates and to issue commands towards the *BLUE-LAND*. In the second phase the attacker breaks into the ICT infrastructure of substation 16 and issues remote commands to start all connected loads. This will lead to a sudden increase in the power demand that cannot be forecasted by automated algorithms. Because operators are completely blinded during the attack, they cannot intervene to start additional back-up generators. Consequently, the disturbance propagates to substations in *BLUE-LAND*, making voltages drop below their normal operating limit and cutting power from railway segments.

The scenario was implemented with the help of the AMICI framework and was tested within the Joint Research Centre's Experimental Platform for Internet Contingencies laboratory. The railway and power grid models were run by two separate *Sim* units and they exchanged data related to consumed power and voltage levels, as shown in Fig. 3. Operator decision units were also implemented as two separate *Sim* units. The experiment used the Modbus/TCP industrial protocol to transfer data between *Sim* units and a pair of *Proxy* units to map between RPC/TCP⟷Modbus/TCP messages for each region. The attacker code that increases the load at substation 16 was implemented as LCC code within a *Proxy* unit. The DoS attack was emulated by turning OFF network interfaces on the hosts running the *Proxy* units.

3.3 Experiment Execution and Analysis of Results

In a first step, the experiment architecture, including networks, PCs and OS, was described through an NS script. This was processed by Emulab that automatically allocated the required resources, it configured VLANs and IP addresses, and it loaded the OSs. Next, we configured the simulators and software components and launched the attack. The experiment employed real Modbus protocols, together with real OS software and real hardware to create a realistic ICT environment.



(a) (b)

Fig. 5: Normal operation: (a) Power Grid, and (b) Railway System

Under normal operation the railway system is powered from the grid and operators can monitor and control in real time the two CIs. As shown in Fig. 5 (a), the level of voltages is directly influenced by the status of trains, i.e. running/stopped, that need to stop at each station and then start off again. Each time a train stops the power drawn from the grid drops to 0MW and increases back after it is started. A change in the load, i.e. in the status of trains shown in Fig. 5 (b), leads to small voltage fluctuations that do not affect the stability of the grid, but can be clearly seen in Fig. 5 (a).

Next, the attack is started on substation 16, where the attacker manages to start-up large consumers and to increase the load to 85MW. Due to interconnections and power flow properties of the power grid, the disturbance propagates to other three substations, i.e. 18, 19 and 20, that are responsible for powering trains in *BLUE-LAND* (see Fig. 6). Here, voltages drop below the operating limit of 0.95 p.u., causing a stop of trains powered by these substations. This clearly shows the side-effects behind strongly interconnected and interdependent systems such as the power grid together with the railway system. Furthermore, it also shows that the attacker does not need to take over substations directly powering train lines, but he can rely on physical properties and the propagation of disturbances to accomplish his goals.

This effect is also shown in Fig. 7, where the start of the attack is marked with *S1*. As power grid operators are completely blinded and unaware of the status of the grid in *BLUE-LAND*, they cannot take additional measures to power the stopped trains. As trains stop, the power consumption drops to 0MW, that is equivalent to the disconnection of several large consumers from the grid. Consequently, voltages increase above the normal operating limit (*S2*). At this point railway operators try to start-up trains again (*S3*), but this crashes voltages and trains stop again (*S4*).

Until this point we have seen the direct dependencies between the three CIs. We have seen that the railway depends on the power grid, but the power grid

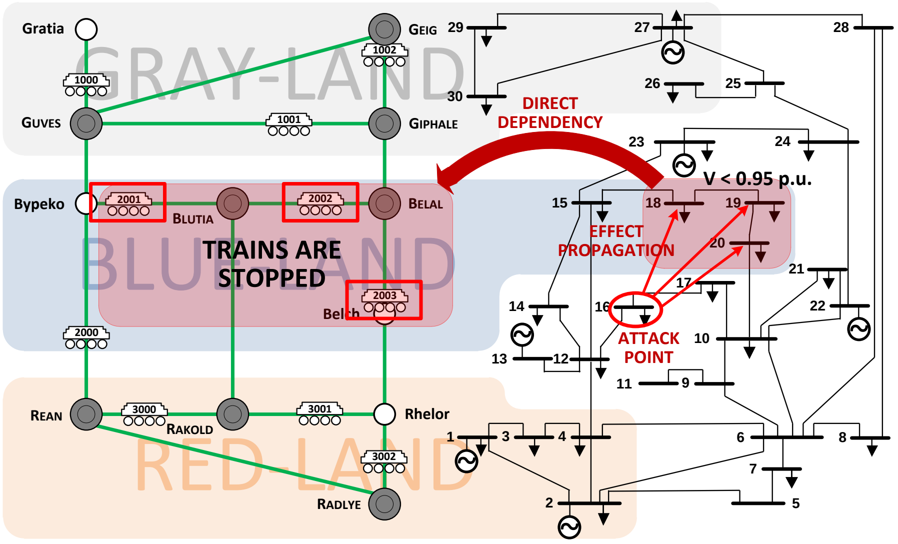


Fig. 6: Propagation of the effect of the cyber attack from the Power Grid to the Railway System

also depends on its ICT infrastructure to ensure normal operation. Without it, voltages drop below operating limits, leaving other critical infrastructures, i.e. railway, without power. However, if power grid operators would be able to realize that their physical infrastructure is under attack, they could take appropriate measures, such as turning ON back-up generators or isolating the substation that caused the perturbation. In our scenario we implemented this aspect by stopping the DoS attack, i.e. by re-enabling network interfaces, which has lead control algorithms to execute for the *BLUE-LAND* and inject an additional of 90 MVars into the grid. The effect can be seen in Fig. 7 at *S5*, where we notice an increase in the level of voltages. This is followed by a restart of trains at *S6*, that this time keeps voltages above their normal operating limit.

To conclude, the scenario presented in this section clearly showed the applicability of AMICI in security studies involving multiple CIs. The actual study performed on three CIs also confirmed the fact that the ICT infrastructure needs to be intact in order to ensure the stability and normal operation of CIs. Furthermore, as CIs get more interconnected and interdependent, there will be a special need of platforms as the one proposed in this paper to analyze these systems.

4 Conclusions

This paper presented AMICI, a novel experimentation platform for analyzing/assessing multiple interdependent Critical Infrastructures. The platform ex-

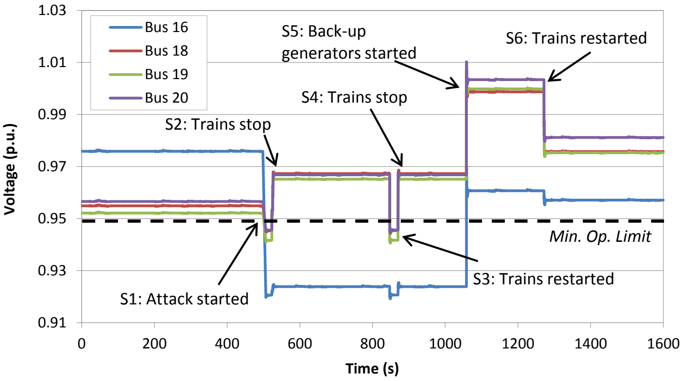


Fig. 7: Scenario execution and effects on power grid voltages

tends our previous work in the field of cyber-physical security experimentation with software components in order to enable a multi-domain experimentation that provides users with functionalities missing from other related approaches: (i) simple integration and inter-connection of multiple CI simulators; (ii) support experimentation with real software and malware in a safe environment; (iii) provides software units that recreate ICT software typically used in monitoring/control of CIs, e.g. SCADA servers, Modbus protocol; and (iv) include automated experiment management capabilities together with a multi-user support. The applicability of AMICI was demonstrated by studying the propagation of perturbations from the ICT infrastructure to a power grid and then to a railway system. The scenario showed that today's CIs are highly interconnected and their normal operation depends on the ICT infrastructure as well as on operator's reactions to contingencies. As future work we intend to apply AMICI to study even more complex systems and interdependencies, with a special focus on ICT infrastructures that can play a crutial role in the outcome of cyber attacks.

Acknowledgments. The authors would like to thank Dr. Roberto Filippini for the very helpful discussions on (inter)dependencies between CIs.

References

1. Bobbio, A., Bonanni, G., Ciancamerla, E., Clemente, R., Iacomini, A., Minichino, M., Scarlatti, A., Terruggia, R., Zendri, E.: Unavailability of critical scada communication links interconnecting a power grid and a telco network. Reliability Engineering & System Safety **95**(12) (2010) 1345 – 1357
2. Falliere, N., Murchu, L.O., Chien, E.: W32.Stuxnet Dossier. `http://www.wired.com/images_blogs/threatlevel/2010/11/w32_stuxnet_dossier.pdf` (2010) [Online; accessed November 2011].
3. McElroy, D., Williams, C.: Flame: world's most complex computer virus exposed. `http://www.telegraph.co.uk/news/worldnews/middleeast/iran/`

9295938/Flame-worlds-most-complex-computer-virus-exposed.html# (2012) [Online; accessed June 2012].

4. Rinaldi, S., Peerenboom, J., Kelly, T.: Identifying, understanding, and analyzing critical infrastructure interdependencies. Control Systems, IEEE **21**(6) (dec 2001) 11 –25
5. Svendsen, N.K., Wolthusen, S.D.: An analysis of cyclical interdependencies in critical infrastructures. In: CRITIS. (2007) 25–36
6. Di Giorgio, A., Liberati, F.: Interdependency modeling and analysis of critical infrastructures based on dynamic bayesian networks. In: Control Automation (MED), 2011 19th Mediterranean Conference on. (june 2011) 791–797
7. Chertov, R., Fahmy, S., Shroff, N.B.: Fidelity of network simulation and emulation: A case study of tcp-targeted denial of service attacks. ACM Trans. Model. Comput. Simul. **19**(1) (2009) 4:1–4:29
8. Davis, C., Tate, J., Okhravi, H., Grier, C., Overbye, T., Nicol, D.: SCADA cyber security testbed development. In: Power Symposium, 2006. NAPS 2006. 38th North American. (2006) 483–488
9. Hopkinson, K., Wang, X., Giovanini, R., Thorp, J., Birman, K., Coury, D.: Epochs: a platform for agent-based electric power and communication simulation built from commercial off-the-shelf components. Power Systems, IEEE Transactions on **21**(2) (2006) 548 – 558
10. McDonald, M., Conrad, G., Service, T., Cassidy, R.: Cyber effects analysis using VCSE. Technical Report, SAND2008-5954, Sandia National Laboratories (2008)
11. Queiroz, C., Mahmood, A., Hu, J., Tari, Z., Yu, X.: Building a SCADA security testbed. In: Proc. of the 2009 Third International Conference on Network and System Security. (2009) 357–364
12. Chabukswar, R., Sinopoli, B., Karsai, B., Giani, A., Neema, H., Davis, A.: Simulation of network attacks on SCADA systems. In: 1st Workshop on Secure Control Systems, Cyber Physical Systems Week. (2010)
13. Mirkovic, J., Benzel, T., Faber, T., Braden, R., Wroclawski, J., Schwab, S.: The DETER project: Advancing the science of cyber security experimentation and test. In: Proc. of the IEEE International Conference on Technologies for Homeland Security (HST). (2010) 1–7
14. Genge, B., Siaterlis, C., Fovino, I.N., Masera, M.: A cyber-physical experimentation environment for the security analysis of networked industrial control systems. Computers & Electrical Engineering (0) (2012) –
15. Genge, B., Siaterlis, C., Hohenadel, M.: On the impact of network infrastructure parameters to the effectiveness of cyber attacks against industrial control systems. International Journal of Computers, Communications & Control **7**(4) (2012) 673–686
16. 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 5th Symposium on Operating Systems Design and Implementation. (2002) 255–270
17. Siaterlis, C., Garcia, A., Genge, B.: On the use of Emulab testbeds for scientifically rigorous experiments. IEEE Communications Surveys and Tutorials **PP**(99) (2012) 1–14
18. Siaterlis, C., Masera, M.: A survey of software tools for the creation of networked testbeds. International Journal On Advances in Security **3**(2) (2010) 1–12
19. –: Zabbix. http://www.zabbix.com/ (2012) [Online; accessed June 2012].
20. Ríos, M.A., Ramos, G.: Power system modelling for urban massive transportation systems. Infrastructure Design, Signalling and Security in Railway (2012) 179–202