Approach to security assessment of critical infrastructures' information systems

R. Leszczyna[1] *I.N. Fovino*[2] *M. Masera*[3]

[1]*Faculty of Management and Economics, Gdansk University of Technology, Narutowicza 11/12, Gdańsk, Poland*
[2]*Global Cyber Security Center, Viale Europa 175, Roma, Italy*
[3]*European Commission Joint Research Centre of the Institute for Energy, 1755 ZG Petten, The Netherlands*
E-mail: rafal.leszczyna@pg.gda.pl; rafal.leszczyna@gmail.com

Abstract: This study presents an approach to the security assessment of the information systems of critical infrastructures. The approach is based on the faithful reconstruction of the evaluated information system in a computer security laboratory followed by simulations of possible threats against the system. The evidence collected during the experiments, stored and organised using a proprietary system InSAW, may later be used for the creation of trust cases which provide valuable information for the end users of the infrastructure. Another new proposal is MAlSim – Mobile agent-based simulator of malicious software (viruses, worms, etc). To the best of the authors' knowledge, such a simulator has not been proposed before. The present approach was applied to the verification of the security of industrial control systems and power plants. In the study, one of the experiments related to the security study of an information system of a power plant, a simulation of zero-day worm attack, is described.

1 Introduction

Critical infrastructures consist of the physical and information technology facilities, networks, services and assets which, if disrupted or destroyed, would have a serious impact on the health, safety, security or economic well-being of citizens or the effective functioning of governments [1].

At the heart of nearly each of these critical infrastructures there is a *process control system (PCS)* [2].

Over the last decade, PCS have passed through a significant transformation. From proprietary, isolated systems to open architectures and standard technologies highly interconnected with other corporate networks and the Internet. Today PCS products are mostly based on standard embedded systems platforms, applied in various devices, such as routers or cable modems, and they often use commercial off-the-shelf software. All this has resulted in reduction of costs, ease of use (thus – less training and increased overall productivity), and enabled the remote control and monitoring from various locations. However, an important drawback derived from the connection to intranets and communication networks is the increased vulnerability to computer network-based attacks.

The number, speed and complexity of network attacks continue to grow. At the same time, PCS are required to provide high reliability, real-time or near-real-time response, minimal operator intervention and automated process changes as some processes are too complex for human resolution in a timely manner.

There is strong need for the assurance that, in addition to the intended operation, PCS will not induce failures or facilitate the intrusion of malicious agents (e.g. hackers and virus). In this context, until recently, information and communication security analyses were concentrated on internal causes (technical components and human operators), and almost exclusively on accidental faults. The increasing use of public information networks requires the systematic consideration of deliberate threats, and as a consequence a more comprehensive view of security encompassing all relevant elements (organisational, technical etc.). The new risks that can derive from the potential violation of the integrity, confidentiality and availability of information, need to be analysed for ensuring proper countermeasures. There is an urgent need for a systematic vulnerability assessment methodology that can provide the assurance of reliable and secure operation of critical networked infrastructures [2].

This article presents our approach to the security assessment of information systems of critical infrastructures (Section 3). The approach is based on the thorough and faithful reconstruction of the evaluated information system in our computer security laboratory (described in Section 4), followed by simulations of possible threats against the system.

During the experiments we collect the evidence and store it and organise using our proprietary system InSAW which facilitates the security analysis performed in the four phases: system description, vulnerability assessment, threat assessment and attack assessment (see Section 6).

The gathered material may be later used for the creation of trust cases, which provide valuable information for the end users of the infrastructure (whether to trust the infrastructure or not, see Section 8). As a result, after the analyses, the operator of the critical infrastructure receives a

documentation which states and justifies the security level of the systems and clearly indicates any present vulnerability. The analysis also gives indications about possible countermeasures against the identified threats.

Another new proposal related to the developed assessment methodology, is the creation of a simulator of malicious software (see Section 7). To the best of our knowledge, such a simulator, which allows simulating viruses, worms etc. in any arbitrary system, is an original piece of work. The simulator plays a very important role in our experiments as malware threats are frequent in the Internet [3].

We have applied our approach to the verification of the security of an existent, fully operative combined cycle electric power plant. One of our experiments – simulation of a zero-day worm attack – is described in Section 5.

2 Related works

Security risk assessment and management of critical industrial IT infrastructures is a relatively new discipline. Most assessment efforts are typically concentrated on corporate information systems. The sub-committee 27 of the first Joint Technical Committee of the International Organisation for Standardisation (ISO) and the International Electrotechnical Commission dedicated the 27000 family standards [4] to this context. In particular:

- ISO 27001 replacing the old BS7799-2 standard defines the Information Security Management System – ISMS.
- ISO 27002 is a code of practice for information security (incorporating the old ISO17799).
- ISO 27003 provides guidance for the implementation of an ISMS.
- ISO 27004 introduces measurements and metrics for security.
- ISO 27005 defines the guidelines for the Information Security Risk Management.
- ISO 27006 defines a set of guidelines for the accreditation process.

While the ISO 27000 family is highly applicable in the context of corporate information systems, it has only few points of reference to the experimental work we are presenting in this paper. More domain specific is the report Q1 published by NIST [5]. In this report, the relevance of the cyber security topic in the context of smart grids is well recognised, and the guidelines for the performance of a Q1 bottom-up ICT security analysis of smart grids are provided. Another set of security guidelines for Industrial Automation and Control systems are described in ISA99 [6].

In order to provide adequate results, a security assessment needs to be fed with all possible data regarding the reliability and security features of system components and the potential threats that might affect them. In context of information systems of critical infrastructures this often means that the validity of security assessments relies heavily on the availability of empirical data, resulting from the observation of the dependability of the target system (or similar systems) under different conditions and states (e.g. normal, under attack, during maintenance etc.). These data are normally gathered through the field observations (e.g. data about vulnerabilities and attack processes collected in real systems).

While systematic experiments are a standard component of many scientific and technical disciplines, including safety engineering, in the security field and especially in the ICT area, applying a systematic, rigorous and methodical approach to experiments is not a common practice. In fact, the ICT security assessment activities, such as penetration testing, red teaming and different so-called ethical hacking procedures, tend to be spontaneous and ad hoc.

In the scientific literature, to our knowledge, only few works try to address this issue: (a) Hussain [7], presents an experiment methodology conceived for the analysis of Distributed Denial of Services; while (b) Herzong [8] presents an Open Source Security Testing Methodology. The first work is mainly dedicated to the particular problem of the denial of services, and for this reason cannot be taken as exhaustive example for experimental security methodologies. The work of Herzong, on the other hand, is mainly a 'guide' for systematic penetration testing, and as it does not take into account aspects as the collection of the experimental data, their aggregation, the definition of the experimental environment etc., it cannot be considered an example of experimental security methodology.

The need of such systematic methodology becomes more than evident looking at the scientific literature in the field of ICT security of industrial critical infrastructures. Creery and Byres [9] presented an interesting high-level analysis of the possible threats to a power plant system, a categorisation of the typical hardware devices involved and some high-level discussion about the intrinsic vulnerabilities of common power plant architectures. A more detailed work on the topic of Critical Infrastructure security is presented by Chandia, Gonzalez, Kilpatrick, Papa and Shenoi [10]. The author shows that communication protocols used in such systems (e.g. Modbus, DNP3 etc.) were not conceived for dealing with typical ICT threats. This is owing to the fact that when they were designed, the world of industrial control systems was completely isolated from public networks, and then ICT-based intrusion scenarios were considered completely negligible. Some work has been done regarding the security of such specialised communication protocols (e.g. [11, 12]).

The issue, which has become very important recently, is the Internet interconnection of the information systems of critical infrastructures. The Internet connections were introduced to critical infrastructures to facilitate their operation and the management, primarily in the administrative departments, but at the same time resulted in the higher exposure of the infrastructures' control systems to the ICT threats common in the Internet. Even if only few critical infrastructures allow for the direct Internet access from the process control system, and most of them separate the system from the other subnetworks 'logically'(we use the term 'logical' separation to distinguish from the 'physical' – being the real disconnection from the other networks) (usually by firewalls), the risk of a security incident remains high. The recent discovery of the Stuxnet malware [13], a cyber worm able to infect process servers and Programmable Logic Controllers (PLCs), and provided with several malicious features such as coordination, automatic update, field device control etc., raised the level of attention on the cyber threats against critical infrastructures at maximum level. This is a new problem facing the area of the critical infrastructures protection and it requires comprehensive study.

3 Critical infrastructures security evaluation approach

Our approach for the security evaluation of the ICT systems of critical infrastructures is based on the simulation of

doi: 10.1049/iet-ifs.2010.0261

attacks against the evaluated systems. To avoid *any* interference with the systems, the experiments are performed in the secure and isolated setting of our computer security laboratory.

The approach comprises the following phases:

- Analysis of the ICT system of the critical infrastructure.
- Reconstruction of the ICT system in the simulation environment.
- Identification of use scenarios.
- Design of experiments.
- Performance of experiments.
- Collection and analysis of results.

In the following, we provide a brief description of each phase.

3.1 ICT system analysis

This phase aims at obtaining a complete 'map' of the ICT system of the critical infrastructure to later reconstruct the system using the hardware and software resources of our laboratory. We study the available documentation of the ICT system and where we encounter any lack of information, we formulate questions to the system administrators, designers and operators. We visit the site, interview the administrators and review the system settings.

The following assumptions are taken:

- The systems and their operational context are known, as well as all the stakeholders, their technical and organisational viewpoints and the processes of different kinds that occur among the different technical systems and actors.
- The system under analysis appertains to an organisation that has defined a complete or partial security policy.

The whole process begins with a thorough system description performed through a fragmentation process. The main objective of this stage is to identify and collect information related to the elements composing the system, based on their relevance under a security perspective. The main idea is to describe the system in terms of components, subsystems, assets etc. The glue connecting together these elements are (a) the concept of service and (b) the concept of data flow. Speaking of services, in the used approach, each component taken into consideration provides to other components or subsystems a set of services, and uses services provided by others in order to fulfil its duties. Considering instead the data flows, they magnify the 'information' interdependence among the elements of the system. A similar systematic approach helps in eviscerating the hidden peculiarities of the target system. For the sake of completeness, here we underline that, as for every security and risk assessment, the quality of the model and of the information used to describe the system is crucial for reaching the completeness and adequacy of the evaluation. If the abstract model provided is weak, the results will be poor. Speaking of complex systems such as those taken into consideration in this paper, it is obvious that the amount of information to deal with might be very excessive resulting in some unwanted disparities in the system representation. While we cannot guarantee in any way the completeness and correctness of the model developed with our technique (and it is hardly provable with any other methodology), we developed a software platform named InSAW (Industrial Security Assessment Workbench) [14], which facilitates the modelling by providing means to achieve high completeness and adequacy via visual descriptions of all elements of the modelled systems, automatic explorations of the generated graphs in search for dependencies, vulnerabilities, cyber-threats and automatic generations of sets of threat scenarios and possible countermeasures. More details about the formal definition of the ICT system analysis can be found in [15].

3.2 Reconstruction of the ICT system in the simulation environment

Having the 'map' obtained in the previous phase, we build a copy of the critical infrastructure ICT system in our laboratory. In this step we have to deal with the limitations of the available resources by making decisions as to which parts of the original system should be reflected completely and which subsystems can be approximated.

The best approach for solving the trade-off between 'in field experiments' and laboratory simulations, is a protected environment composed of the following elements:

- A *Production System* containing the most significant elements of the system under analysis, and aiming at recreating as detailed as possible all the typical profiles, data flows and states under analysis.
- A *Horizontal Service Area* providing all the services needed for the maintenance of the laboratory. Example of such services could be *backup service* (for creating and storing the image of every ICT component required for the experiment), *Interconnection Service* (providing the connection with the external world to allow a fast retrieval of patches, configuration and security information), *Network Configuration Service* (for recreating in a centralised and automatic way all the possible network architectures needed for implementing different test scenarios) etc.
- An *Attack System* enabling the simulation of different kind of attack configurations and scenarios.
- An *Analysis System* containing a set of analytic engines for pre- and post-experimental analysis.

It is evident that the most important part of this protected environment is related to the simulation of the production system. The study of complex systems, either physical or cyber, could be carried out by experimenting with real systems, software simulators or emulators. Experimentation with real production systems suffers from the inability to control the experiment environment to reproduce results. Furthermore, if a study (as in our case) intends to test the resilience or security of a system, there are obvious concerns about the potential side effects (faults and disruptions) to mission critical services. On the other hand, the development of a dedicated experimentation infrastructure with real components is often economically prohibitive although the disruptive experiments on top of it could constitute a risk to safety. Software-based simulation has always been considered an efficient approach to study physical systems, mainly because it can offer low-cost, fast and accurate analysis. Nevertheless, it has limited applicability in the context of cyber security owing to the diversity and complexity of computer networks. Software simulators can effectively model normal operations, but fail to capture the way computer systems fail. Based on these facts, we have chosen to follow a hybrid approach in between the two extremes of pure simulation and

experimentation with only real components. In [16], we proposed a framework that uses simulation for the physical components and an emulation test bed based on Emulab [17] to recreate the cyber part of PCS, for example, SCADA servers, corporate network etc. The models of the physical systems are developed in Matlab Simulink from which we generate the corresponding 'C' code using Matlab Real Time Workshop. The generated code is then executed in real time and is able to interact with the real components of our emulation test bed. Using this experimental framework, as showed in [16], we have been able to accurately recreate large PCS, for example, having up to a hundred PLCs. Moreover, in the particular case of the experiments described in this paper, we have also taken advantage of a real production system, described in Section 4, integrated with our simulated devices.

3.3 Identification of use scenarios

We analyse how the ICT system is used, which users access it, what are their rights and the operational space. Then we document it as use scenarios. This phase is extremely important since, if well done, it allows capturing in deep the mechanisms governing the infrastructure under analysis. In particular, focus is on all information related to users, rights, operational procedures, security policies, system states and related procedures (e.g. maintenance state etc.).

The knowledge obtained at this stage is integrated with the information gathered during system analysis, using as glue the concept of dependency and, again, of service. Roughly speaking all this knowledge is organised as in a big oriented n-dimensional graph composed by different classes of nodes (components, users, stakeholders, subsystems etc.) and linked by means of services, dependencies and data flows (the graph edges). These graph edges can be simple or weighted where the weight might represent information like the 'minimum level of QoS to be guaranteed', the relevance for the entire system, the economic value etc.

The identification of use scenarios, allows one to 'project' the global static description of the system under analysis on specific subsets representing the relevant elements to be taken into consideration for the experimental phase. Details on the projection phase can be found in [18].

3.4 Design of experiments

When designing the experiments, we first define the attack goals and the system sections that will be affected. Then we describe the attack scenarios depicting subsequent steps required for the successful attack. These textual descriptions are accompanied with more formal attack specifications by means of a particular type of multi-dimensional attack trees [18]. Those attack trees are conceived to be easily integrated with the multi-dimensional graph-based system representation described in the previous sections. Finally, we define the system conditions for the successful performance of the attacks and experiments (such as environmental settings, the required resources etc).

3.5 Performance of experiments

Before each experiment we make sure that the simulation environment is in 'zero-state' – the initial state defined in the corresponding use scenario. Then the image of the system settings is created to make the experiments repeatable. Our experimental environment [16] was designed and configured so that it is possible to completely automatise all the phases of the experiments, through scripts, as well as to store automatically in an experiment library all the settings related to the initial state of the system (owing to the use of the Emulab platform). In this way, every run of the experiment can be guaranteed identical to the precedent in terms of initial conditions, environmental parameters and triggering events. After that we start performing the experiment. A set of network and host-based sensors have been introduced to gather a wide set of information about the behaviour of the system during the run of the experiment. Examples of the information gathered are, among the others:

- Values and set point trends of the simulated physical installation (temperature evolutions, pressure evolution etc.).
- Control traffic exchanged between SCADA servers and PLCs.
- Load information related to the control servers.
- Logs coming from ad-hoc configured Intrusion Detection Systems and Firewalls.

In this way, the system events are recorded. The granularity of the event logging depends obviously on the number and type of rules (expressed using first-order logic expressions and IDS style rules) defined by the analyst running the experiments.

3.6 Collection and analysis of results

In the final phase, we collect the information about the system events. We process it to extract the key, attack related, information. We analyse the information and formulate conclusions about the security of the ICT system. Where system vulnerabilities are discovered, we propose countermeasures.

The approach based on the simulation of the attacks in the reconstructed environment of the evaluated system facilitates the identification of the vulnerabilities and the countermeasures in comparison with an analytical approach where the system architecture, configuration and its performance must be thoroughly analysed. The vulnerabilities are discovered in the real-time basis, 'on-the-fly'. It means when a simulated threat agent is able to explore a system vulnerability, the effect is promptly noticed and notified to the analysts. Especially the performance part of the system analysis is difficult to investigate in the analytical manner, as reflecting all the system states requires time and a thorough, systematic approach so as not to lose any aspect of the operation of the system. In the simulation approach, on the contrary, the reconstructed system is running in a natural way, in the mode consistent to the scenarios, and all the states are introduced inherently with the operation of the system. As a result, there is no concern of faithfully reflecting the system states as this feature is provided automatically.

4 Simulation environment

The approach described in this paper has been successfully applied in several contexts. In particular, we used it to assess security of industrial control systems and power plants.

In this section, we provide a brief description of the simulation environment as configured for the application in the power plant domain. A power plant has a quite complex environment, comprising several kinds of systems,

doi: 10.1049/iet-ifs.2010.0261

subsystems and components. Following the approach, in the first step, we analysed the target system and its functions to identify and classify: assets, data flows, components, clusters of components (subsystems), services and dependencies among the different services.

According to our analysis, the system includes several main subsystems (Fig. 1):

- The *Field System*, hosting all the PLC, RTU and sensors of the power plant.
- The *Process Control and Data Acquisition System* (Process SCADA), which basically control the field system.
- The *Control Network*, which provide the communication service among the whole Power Plant.
- The *Data Network*, allowing to interconnect different Power Plants.
- The *Business (Offices) Network* with the typical intranet applications.
- The *Demilitarised Zone* (*DMZ*) where servers for sharing process related data are located.

These systems were reconstructed in the secure isolated environment of our laboratory based on 120 hosts, the network equipment necessary to interconnect them (which includes 16 network switches), as well as SCADA devices set up over physical hydrologic installation – the *Physical Power Plant Emulator* (see Fig. 2 and Table 1). The isolation of the laboratory environment means that the simulation architecture was physically disconnected from any other networks which could eventually be connected to other systems. This isolation was maintained to avoid any possible interferences with other systems even if the simulated threat agents were designed and implemented to be harmless (the malicious payload is always removed from the reconstructed threats).

In this environment, the information system of the power plant was reconstructed with very high fidelity. The identical subnetworks were created. All the key workstations of the power plant were copied in one-to-one relation. It means each of the workstations was reflected into one host of the simulation environment. Only stations

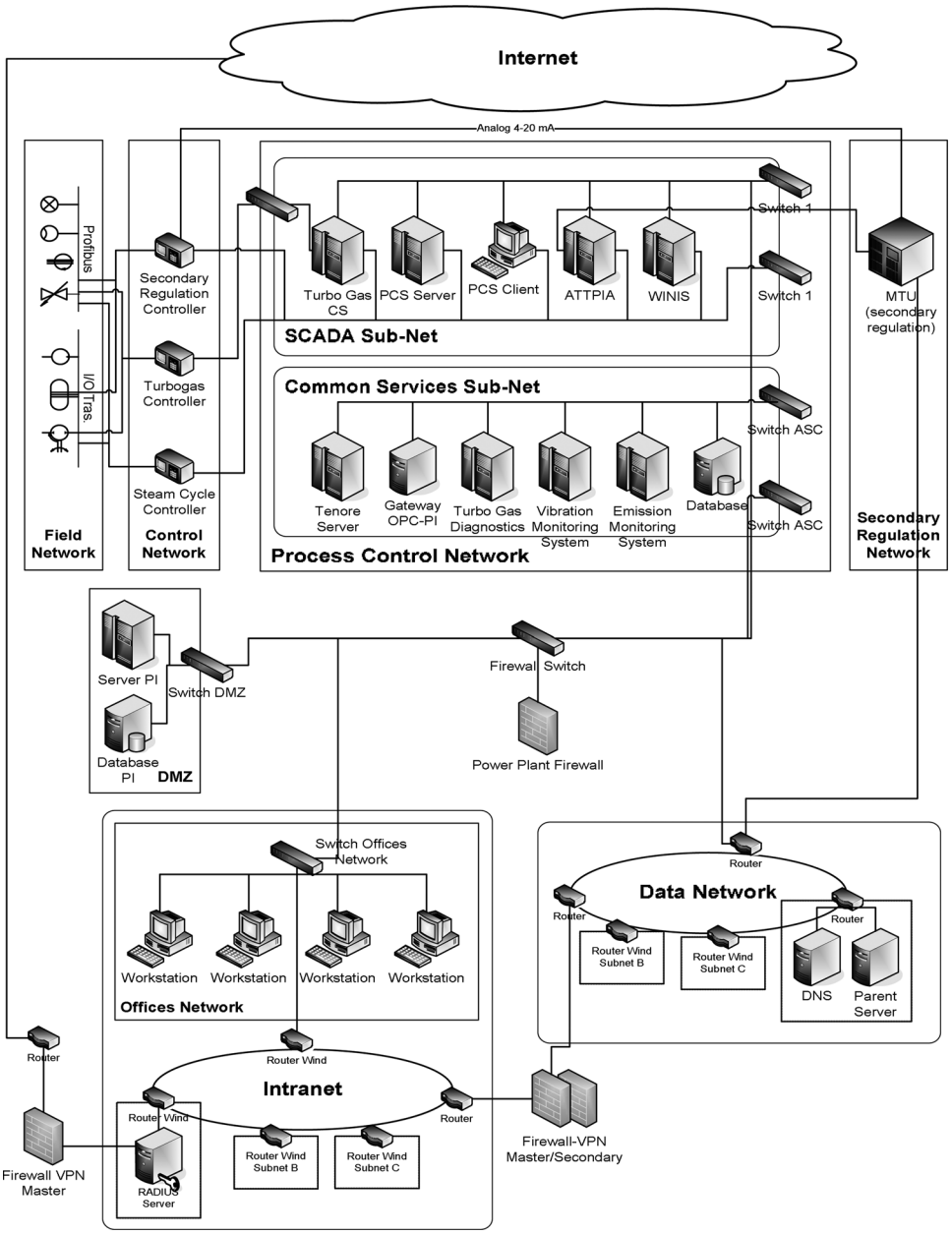

Fig. 1 *Reconstructed information system of a power plant*

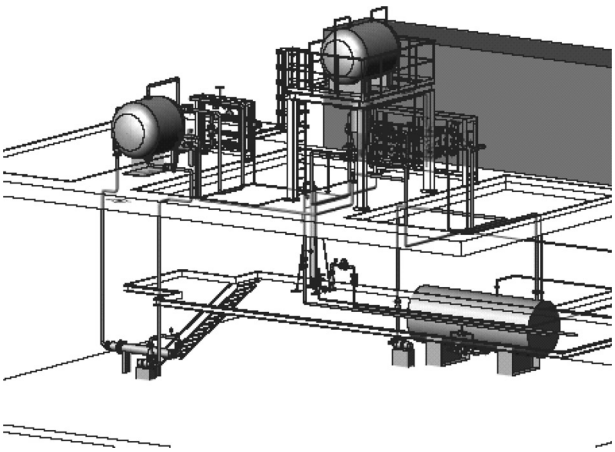

Fig. 2 *Physical power plant emulator*

of the Intranet were approximated with a lower number of hosts, but this was without loss of generality. In the reconstruction, the same network addresses were used, the same software installed (including the level of patching), the same configurations of firewalls applied etc.

Additionally the environment comprises the auxiliary parts that support the configuration, performance and observation of the experiments or provide any other auxiliary functionality:

1. *Threat and Attack Simulator*, which aims at providing conditions for reconstructing attacks and threats that can jeopardise the analysed information system. This is the part of the simulation environment where the simulated attacks are configured and launched. Since there are various and diverse attacks, when designing this part of the simulation environment, we pay attention to assuring high flexibility and easiness of configuration. The Threat and Attack Simulator allows managing virtual subnetworks and creating multiple virtual network nodes. These, together with the hosts, are easily configurable and provided with diverse resources. Particularly, they include various software, that is, operating systems and the specialised programmes for developing attacker tools and for performing the attacks.
2. *Observer Terminal*, which is used for monitoring the traffic of the Mirrored Information System in order to evaluate the effects caused by the simulated attacks on the system. It tracks all the malicious or anomalous events happening in the Mirrored Information System during the tests and experiments, and stores them in the central database.
3. *Vulnerabilities and Countermeasures Repository*, where we store all information about system vulnerabilities and the relative countermeasures. It is composed of two subsystems: the Vulnerabilities and Countermeasures Database and the Binaries Repository. In the former, we store knowledge about existing and known vulnerabilities, threats, attacks and countermeasures, while the latter is devoted to storing and cataloguing attack tools, such as packet generators, Trojan horses and root-kits, and other executable code to be used in security experiments carried out in the simulation environment. The Vulnerabilities and Countermeasures Repository is implemented within the InSAW framework (see Section 11). Q2
4. *Testbed Master Administrator*, used to remotely manage both the network and the experiments. It manages the operations related to the initiation and termination of experiments and allows real-time observation of the behaviour of each system during simulations.
5. *Horizontal Services*, responsible for providing services that are needed for the efficient management of the simulation environment such as backup services or file-sharing services.

Further details about the simulation environment can be found in [19, 20]. Q3

5 Case study: simulation of zero-day worm attack

We used the simulation environment to assess security of a power plant infrastructure. An existent, fully operative combined cycle electric power plant was reconstructed and evaluated during the experiments.

In order to perform this evaluation, we reconstructed the network setting of the power plant, that is, we emulated the following subsystems (Fig. 1):

- *Process Control Network*, which interconnects diverse subsystems of the energy production process.
- *Field Network*, which links controllers and field devices.
- *Data Network*, where power production process related data are archived.
- The corporate network (*Intranet*).
- *DMZ*, where power generation process can be monitored from outside.

In this setting, we performed the simulation of a *zero-day* worm attack. A zero-day (or zero-hour) attack is a computer threat that exposes undisclosed or unpatched computer

Table 1 Components of the physical power plant emulator

Siemens	Emerson	ABB	Field dev.
2 × OpenPMC (PLC)	2 × Ctrl MD (PLC)	2 × AC 800F (PLC)	21 PA
2 × IM157 (DP Link)	1 × KLD-2 (DP/PA)	3 × RLM 01 (Y Link, repeater)	DP
2 × DP/ PA Coupler	1 × KLD-2 (DP/PA)	1 × Converter F.O./RJ45,	FF
2 × ET 200M (active bus) 1 × SM321 (DI)		Ethernet	3 Hart
1 × SM322 (DO)		1 × Switch Ethernet	12 analog I/O
2 × SM331 (AI)		2 × CI 840	
2 × SM332 (AO)		1 × RLM 01	
		1 × DP/PA Power Link	
		2 × LD 800 HSE	
		1 × Converter F.O./RJ45, Ethernet	
		1 × Switch Ethernet	

doi: 10.1049/iet-ifs.2010.0261

application vulnerabilities. Zero-day attacks take advantage of computer security holes for which no solution is currently available. Zero-day exploits are released before the vendor patch is released to the public. A zero-day exploit is usually unknown to the public and to the product vendor.

We have developed the following attack scenario:

A power plant operator working on a PC located in the power plant's Intranet browses the Internet and is accidentally infected by a worm that has been just launched in the recent hours. This is a new type of worm, not just a slight modification of an existing one. For this reason, and because of the fact that the worm is so recent, it is yet unknown to the antivirus community (zero-day worm). Its signature is not stored in any antivirus database.

The worm infects programmes on the user's PC and, taking advantage of the fact that unlimited traffic between the hosts in the Intranet is allowed, it infects also the remaining hosts of the Intranet. Later on the user, unconscious of the fact that his/her PC is infected by the worm, opens a VPN connection to a host in the Process Control network. Now the worm has a free passageway to the critical subnetwork of the power plant network. It moves through it and starts infecting the computers in the Process Control network. Simultaneously, the adverse effects of the worm begin to be apparent. The computers become less effective, the applications raise errors and stop functioning, and the network connections are lost.

The general aim of worm attacks is to infect as many computers in the Internet as possible and to cause their malfunctioning. The attack is not particularly oriented against power plant systems. However, when reaching the network of the power plant, the worm can reach the Process Control Network and Intranet subsystems and cause severe damage.

In the simulation, the worm, simulated by MAlSim gradually infected all hosts in the Intranet and progressively in the Process Network, starting from the PCS Server. After this propagation wave, the worm copies remained in all the hosts through which it passed, were deactivating the hosts' network cards, and making any network-related operation impossible.

As a result, the following services were affected:

- *Power Generation Control*, which controls and monitors the power production process. The viral infection and the consequent loss of connection with the direct controllers of the power generation devices, made impossible controlling the power production process from the Process Network. The operators were forced to use older, low-level control infrastructure.
- *Power Generation Data Acquisition*, which provides the information necessary for the power plant supervision and for production planning. In the interval between the worm outbreak and the system recovery, the data could not be collected. The operators were forced to use the alternative low-level process control and monitoring infrastructure and to make production plans in non-automated manner. The information generated by the service is also delivered to some third-party companies, for whom the interruption in the delivery of the data was alarming.
- *Anomaly Diagnosis*, which monitors and analyse the vibrations of power production devices (primarily – the gas turbine), in order to predict or early detect faults or malfunctions. This service allows, for example, predicting the effects of utilisation regimes of devices, supporting decisions about their maintenance or replacement in at least weeks of advance. Since the full system recovery of the Process Network (based on restoring the last safe system state from backup copies) should not take more than 3 days (at maximum!), the loss of the anomaly diagnosis-related information in the time shall not result in any serious consequences.
- *Gas Exhaust Management*, which provides information on the quality of gas emissions to the atmosphere, to interested third parties (e.g. local authorities). Provision of this service is imposed by law. Without the service, a plant cannot obtain the authorisation for energy production or the continuation of the production. Severity of the threat in regard to this service depends on the particular regulations of the country. In our case, the regulations accept lack of data for, at maximum, a 3-day period (maximal system recovery time, see the previous bullet). In general, restitution of the data with the estimations based on the proceeding and the following periods, and the production plan for the period of the interruption of data delivery, should suffice.
- *Remote Maintenance*, which conducts software patching, updating from Intranet (and eventually the Internet) by authorised actors, including third parties. The impact of the worm in relation to the service is obvious – the software maintainers have to come to the site to remove the effects of the infection.

Summarising, the effects of this particular worm infection, though critical, were not dramatic. The power plant could continue its normal operation – from the point of view of the power production process. The damages were mostly related to the interruption of data delivery, and to the necessity of performing less automated control over the production process.

This is because the payload of the simulated worm aimed at deactivating the network adapters of the infected computers, causing only the loss of connectivity. However, another more malicious version of the worm could, for example, interfere with the communication protocol through which actual commands are sent to the field actuators, for then causing anomalies in the power production process.

To develop such a dedicated worm targeting industrial systems, an advanced level of the recognition of the power plant infrastructure is required, including good knowledge of SCADA protocols. Even more, this new worm will have to spread quickly enough to overpass its signature recognition and detection by malware detection engines.

Finally, it must be noted that it is very difficult to prevent from zero-day attacks, as its strength is based on its urgency and unexpectedness. No signature-based antimalware software will be prepared for the detection of this kind of attack, and will let the malware spread. A possible solution for protection against this type of attacks could be to use anomaly detection-based malware detection engines.

The experiment allowed us to assess the resistance of the power plant information system in the conditions practically identical to real. As described above (Section 4), the simulation environment was deployed over 120 hosts, interconnected with all the necessary network equipment, as well as SCADA devices set up over physical hydrologic installation. There we reconstructed all the power plant subnetworks and made copies of all key workstations, where each workstation was reflected into one host of the simulation environment. Only stations of the Intranet were approximated with a lower number of hosts, but this was

without loss of generality. All the configurations were made as in the original system. In our opinion, simulating such a complex environment, with its all hardware, software and settings, based solely on a modelling software is impossible. This approach would require applying simplifications to model the simulated environment.

On the other hand, precise conceptual analysis based on the software and hardware specifications and the knowledge of the evaluated system would require much more time in comparison with the setting and performance of simulations. It also might be too complex for completing successfully. In our approach we do not need to know all the internal settings of software as we use images of the memory storages of the reconstructed stations. Then we just run an experiment and observe its results. As far as the simulation of the zero-day worm attack some effects might seem to be obvious (for example, that the malware, being 'zero-day', will pass the antivirus security) but until verified one cannot be certain that he/she would predict all the effects. For example, the firewalls of the system might not let the worm to pass to another network zone. We can also observe, in real time, the behaviour of the attacked systems and the already installed security solutions such as firewalls or antivirus tools. The timing, the effects etc. Q4

Additionally performing the simulation provided us with an empirical evidence of the power plant system behaviour in face of a zero-day worm attack, sort of the 'proof' of the attack effects, which for many is more convincing than pure oral or written statements.

Another, very important benefit from applying this type of approach is that we can demonstrate the effects and the course of events. After we had successfully performed our experiments we invited for the demonstration the management of the power plant as well as its regular employees. We performed the simulation showing to all the participants of the demonstration how the system would behave when attacked and which were the effects. After the presentation many questions were raised in relation to the security of the power plant systems, our answer to them, together with the impression made by the demonstration definitely boosted the participants' awareness of information security issues. The general outcome was that on one side, the management was more willing to invest into security solutions and competent security staff while on the other – the workers were more likely to accept the burden imposed by the security restrictions.

6 Industrial security risks assessment workbench

The design of experiments is not a trivial task, and the use of a tool helping to analyse the critical infrastructures and their interconnections in order to identify implicit dependencies, to detect potential cascading effects, and finally to identify vulnerabilities, threats and attacks which could cause major damages, would be more than desirable.

We have chosen to adopt as reference the *Industrial Security Risks Assessment Workbench* (InSAW) presented by Nai and Masera in [14–16, 18]. The methodology proposed by Masera and Nai foresees that in order to assess the security of a system, it is necessary to provide a description of the system itself, of its components, of its assets, of the interaction and the relationships among the components, the assets and the external world. This description (expressed analytically by tables) could be used to identify in a systematic way the vulnerabilities affecting the whole system. Moreover, analysing the vulnerabilities affecting the different components–subsystems–services of the target system, while at the same time inspecting the different relationships–dependencies–data flows linking together all the actors of the system, it is possible to build a graph of disservice chains, that is, a graph that systematically illustrates all the possible explicit and implicit cascading effects which can be caused by a low-level component affected by a vulnerability. These vulnerabilities are then described by some significant parameters (e.g. severity, plausibility, resource costs etc.) and used to identify the threats that can be associated with the relevant services provided by the system. Such information is then used to identify and validate candidate attacks that can be exploited against the system. This evaluation gives as feed-back a set of 'feasible attacks' with associated indexes which show off the level of exposure of the system. All these operations are quantified by some risk-related indexes that are then employed to perform the evaluation of the security failure risk and the countermeasures.

The systematic use of the tool implementing this methodology allows magnifying off-line the most interesting attack scenarios, those which might, in some way, interfere with the most relevant services and assets of the system under analysis.

7 MAlSim

During our studies we have encountered the problem of lack of software and methodology for the simulation of *malware* [the analysis of existing solutions for malware simulation an interested reader may find in [20]; the study made evident that there are no compound frameworks for simulation of malware that would support the security assessments of information systems based on simulation of attacks] – malicious software that run on a computer and make the system behaving in a way wanted by an attacker [21]. Ed Skoudis and Lenny Zeltser [21] as well as Peter Szor [22] proposed classifications of malware, grouping in this family viruses, worms, malicious mobile code, backdoors, Trojan horses, rootkits and combined malware (hybrids). Malicious mobile code is a malicious lightweight programme that is downloaded from a remote system and executed locally with minimal or no user intervention [21]. Skoudis and Zeltser illustrated this type of malware on the example of ActiveX controls, Javascript, VBScript or Java, programmes. Today also electronic documents and multimedia files (PDF, Flash or RealPlayer files etc) which can be executed by a viewer or player may contain malicious code. According to the study of Symantec Security Response team the most popular Internet-based attack for the second quarter of 2010 was related to malicious PDF activity, which accounted for 36% of the total threats (57% in the previous quarter). The study of Symantec shows that malware together with exploits are the most common attacks in the Web [23]. As today most of critical networked infrastructures are connected to the Internet these types of attacks pose a serious threat against them [3]. For answering this issue, we decided to develop a malware simulation tool.

MAlSim – Mobile Agent Malware Simulator is a software toolkit that aims at simulating malicious software in computer network of an arbitrary information system. The framework aims at reflecting the behaviours of various families of

doi: 10.1049/iet-ifs.2010.0261

malware (worms, viruses, malicious mobile code etc.) and various species of malware belonging to the same family (e.g. macro viruses, metamorphic and polymorphic viruses etc.). It can simulate well-known malware (e.g. Code Red, Nimda, SQL Slammer), but it can also simulate generic behaviours (file sharing propagation, e-mail propagation) and non-existent configurations (which supports the experiments aiming at predicting the system behaviour in the face of new malware). MAlSim is a distributed simulator that simulates behaviour of each instance of malware independently. This means that if the prototype malware propagates over a network, making its copies, then the MAlSim agent dedicated to simulate this malware, also spreads across a network and creates new instances of itself.

Further details of MAlSim may be found in [20].

8 Trust case

The results of experiments performed in our simulation environment form the evidence to support the argumentation which we present in *trust cases* in order to provide means to justify trust into a system.

Trustworthiness of IT systems and services is an issue of growing importance, in particular with respect to properties such as safety, privacy and security. Trustworthiness means that there are 'good' reasons for trusting that a given object possesses a distinguished property (or set of properties).

Trust case is the means of communication to convey a message that serves to build user's trust to a system. From the technical point of view *trust case* is a data structure that encompasses argument and related evidence, which together demonstrate that an object (a system, an infrastructure, an organisation) exhibits certain precisely defined properties. Trust case documents are presented in a graphic form, which significantly boosts legibility and helps to maintain soundness of arguments.

A trust case has a tree-like structure and is composed of nodes of different types. The basic node type is *claim*, which contains a concluding statement to be analysed. A node of type *argument* can be linked to the claim and then the corresponding *premises* and *warrant* are linked to the argument node. A premise can be of the following types: an assumption represents a premise that is not further analysed in the trust case; a claim represents a premise to be further analysed by a more detailed argument; and a fact represents a premise that is obviously true or otherwise is supported by some evidence. The evidence is provided in external (to the trust case) documents that are pointed to by nodes of type reference.

There are no criteria of an 'acceptable' case. This issue is to be decided by the trust case user and/or an expert acting on her/his behalf. Instead, we are focusing on general aspects of argumentation and the question of how to build a valid argument based on the available evidence. Consequently, the convincing power of a trust case becomes a subjective issue. Nevertheless, we are still interested in the assessment of the compelling power of a trust case and we work towards providing an appropriate support to this task.

More details about the concept of trust cases an interested reader may find in [24, 25].

9 Conclusions

We have developed an approach that allows us to assess security of critical infrastructures. The approach after being applied to assess the security of multiple critical infrastructures across Europe helps us in obtaining more general relationships and principles concerning security of critical networked infrastructures. Each project (understood as a separate security analysis of subsequent system) brings in very interesting observations regarding the protection level of critical infrastructures' information systems. We can observe trends in applied network topologies, use of firewalls and other security measures, and their configurations, user privileges, applied security levels etc. Based on these observations we are ready to formulate more general security polices applied at various levels: at the infrastructure level, at the regional level, at the national level and at the Union level (as nowadays critical infrastructures has become highly interconnected and they span across the borders [1]).

In the paper, it is argued that experimental security is needed as a basic discipline for the supply of data for the assessment of the security of critical industrial systems, owing to the lack of data originating from the real world. The scarcity of these security data can undermine the attempts to protect systems for the deficiency of adequate understanding of their vulnerabilities and the potential impact of malicious attacks. This approach has already been applied in several industrial contexts (e.g. [12]).

An open question, at the moment, is related to the identification and measurement of the so-called 'security metrics' and 'security parameters'. In other words, which parameters can be used as invariants for quantifying the security of a system and for comparing the results of different security experiments carried out on various systems and architectures? Security is not a functional or structural variable that can be readily measured, and reducing it to the satisfaction of some features such as integrity, availability and confidentiality is not enough. Security is more than an on–off quality, and there is the need to determine whether the security of a system betters or worsens in different scenarios. Further developments of the approach will also refer to its better integration with Trust Cases as well as applying GAM [26] to support the Design of Experiments phase of the security evaluation process. In addition, MAlSim requires additional development [20].

10 Acknowledgments

We would like to thank Janusz Górski, Łukasz Cyra and Aleksander Jarzębowicz as well as the other members of IAG (Information Assurance Group, http://iag.pg.gda.pl/iag/) for their valuable input concerning trust cases.

11 References

1 European Commission: Communication from the Commission to the Council and the European Parliament: Critical Infrastructure Protection in the Fight Against Terrorism. Internet, October 2004
2 Miller, A.: 'Trends in process control systems security', *IEEE Secur. Priv.*, 2005, **0**, pp. 57–60 Q5
3 SecurityFocus: 'SecurityFocus vulnerability database'. http://www.securityfocus.com/bid (last accessed 9 September 2010)
4 ISO 27000 standards, http://www.iso.org
5 The Cyber Security Coordination Task Group, LEE, A., Brewer, T. (Eds.): 'Smart grid cyber security strategy and requirements', Draft NISTIR 7628, September 2009
6 ISA99, http://www.isa.org
7 Hussain, A., Schwab, S., Thomas, R., Fahmy, S., Mirkovic, J.: 'DDoS experiment methodology'. Proc. DETER Community Workshop on Cyber Security Experimentation, June 2006
8 Herzog, P.: 'Open source security testing methodology manual' (Institute for Security and Open Technologies, http://OSSTMM.org

9 Creery, A., Byres, E.: 'Industrial Cyber-security for power system and SCADA networks', *IEEE Ind. Appl.*, 2007, **13**, (4), pp. 49–55
10 Chandia, R., Gonzalez, J., Kilpatrick, T., Papa, M., Shenoi, S.: 'Security strategies for scada networks'. Proc. First Annual IFIP Working Group 11.10 Int. Conf. on Critical Infrastructure Protection, Dartmouth College, Hanover, New Hampshire, USA, 19–21 March 2007
11 Majdalawieh, M., Parisi-Presicce, F., Wijesekera, D.: 'Distributed network protocol security (DNPSec) security framework'. Proc. 21st Annual Computer Security Applications Conf., Tucson, Arizona, 5–9 December 2005
12 Mander, T., Nabhani, F., Wang, L., Cheung, R.: 'Data object based security for DNP3 Over TCP/IP for increased utility commercial aspects security'. Proc. Power Engineering Society General Meeting, Tampa, FL, USA, 24–28 June 2007 (IEEE, Los Alamitos, 2007), pp. 1–8
13 Stuxnet: http://www.symantec.com/connect/blogs/stuxnet-introduces-first-known-rootkit-scada-devices
14 Nai Fovino, I., Masera, M.: 'InSAW-industrial security assessment workbench'. Proc. Int. Conf. on Infrastructure Systems, Rotterdam, 10–12 November 2008
15 Nai Fovino, I., Masera, M., DeCian, A.: 'Integrating cyber attacks within fault trees', *Int. J. Reliab. Eng. Syst. Saf.*, 2009, **94**, (9), pp. 1394–1402
16 Nai Fovino, I., Genge, B., Siaterlis, C., Masera, M.: 'A framework for analyzing cyber-physical attacks on networked industrial control systems'. Fifth IFIP WG 11.10 Int. Conf. on Critical Infrastructure Protection Dartmouth College, Hanover, New Hampshire, USA, 23–25 March 2011
17 Emulab - Network Emulation Testbed – http://www.emulab.net/
18 Nai Fovino, I., Masera, M.: 'Through the description of attacks: A multidimensional view'. Proc. 25th Int. Conf. on Computer Safety, Reliability and Security, Gdansk, Poland, 26–29 September 2006
19 Leszczyna, R., Fovino, I.N., Masera, M.: 'Security evaluation of IT systems underlying critical networked infrastructures'. Proc. First Int. IEEE Conf. on Information Technology (IT 2008), Gdansk, Poland, May 2008
20 Leszczyna, R., Fovino, I.N., Masera, M.: 'Simulating Malware with MAlSim', *J. Comput. Virol.*, 2008, **0**, pp. 0–1. Available at: http://www.springerlink.com/content/k0843hgq60333556 (last accessed 3 May 2010) **Q5**
21 Skoudis, E., Zeltser, L.: 'Malware: fighting malicious code' (Prentice Hall Professional Technical Reference, Upper Saddle River, NJ, USA, 2003)
22 Szor, P.: 'The art of computer virus research and defense' (Addison Wesley Professional, 2005, 1st edn.)
23 Fossi, M., *et al.*: 'Symantec intelligence quarterly april' (Symantec Corporation, June 2010) **Q6**
24 Górski, J., Cyra, Ł., Jarzębowicz, A., Miler, J.: 'Argument strategies and patterns of the trust-IT framework', *Pol. J. Environ. Stud.*, 2008, **17**, (4C), pp. 323–329
25 Górski, J., Jarzębowicz, A., Leszczyna, R., Miler, J., Olszewski, M.: 'Trust case: Justifying trust in an IT solution', *Reliab. Eng. Syst. Saf.*, (Safety, Reliability and Security of Industrial Computer Systems) 2005, **89**, (1), pp. 33–47
26 Cyra, Ł., Górski, J.: 'Extending GQM by argument structures'. Proc. Ninth Natl Software Engineering Conf. (KKIO), Poznań, Poland, October 2007
27 Fovino, I.N., Masera, M., Decian, A.: 'Integration of cyber-attack within fault trees'. In 17th European Safety and Reliability Conf. (ESREL), June 2007, vol. 3, pp. 2571–2578
28 Masera, M., Fovino, I.N.: 'A service oriented approach to the assessment of infrastructure security', in Goetz, E., Shenoi, S. (Eds.): 'Vol. 253 of IFIP international federation for information processing' (Springer, 2008), pp. 367–380

doi: 10.1049/iet-ifs.2010.0261

IFS20100261

Author Queries

R. Leszczyna, I.N. Fovino, M. Masera

Q1 Define NIST, ICT, SCADA, RTU, VPN.

Q2 Please check the citation of Section 11.

Q3 References have been renumbered to arrive at the sequential order of citation. Please check and confirm.

Q4 Please check the sentence 'The timing, the effects etc.' as the sentence is incomplete.

Q5 Please check the volume number and page range in Refs. [2] and [20].

Q6 Please provide next two author names for *et al.* reference [23] as per IET reference style.