Security evaluation of IT systems underlying critical networked infrastructures

Conference Paper · June 2008

DOI: 10.1109/INFTECH.2008.4621614 · Source: IEEE Xplore

3 authors:

Rafal Leszczyna
Gdansk University of Technology
37 PUBLICATIONS **212** CITATIONS

SEE PROFILE

Igor Nai Fovino
European Commission
82 PUBLICATIONS **1,718** CITATIONS

SEE PROFILE

Marcelo Masera
European Commission
124 PUBLICATIONS **1,608** CITATIONS

SEE PROFILE

This is a postprint of the paper presented at the 1st International Conference on Information Technology, IT 2008, 19-21 May 2008, Gdańsk, Poland. DOI: 10.1109/INFTECH.2008.4621614

Security Evaluation of IT Systems Underlying Critical Networked Infrastructures

Rafał Leszczyna, Igor Nai Fovino, Marcelo Masera
European Commission Joint Research Centre
Institute for the Protection and security of the Citizen
Via Enrico Fermi 2749, 21027 Ispra (VA), Italy
rafal.leszczyna@jrc.it, igor.nai@jrc.it, marcelo.masera@jrc.it

Abstract

In recent years, critical infrastructures have become highly dependent on information and communication technology (ICT). The drawback of this situation is that the consequences of disturbances of the underlying ICT networks may be serious as cascading effects can occur. This raises a high demand for security assurance, with a high importance assigned to security evaluations. In this paper we present an experiment-centric approach for the characterisation and assessment of security threats to information systems of industrial critical infrastructures. The description of the approach is followed with a presentation of the supporting hardware and software architecture.

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 [2].

In recent years, critical infrastructures have become highly dependent on information and communication technology (ICT). The drawback of this situation is that the consequences of disturbances of the underlying ICT networks may be serious as cascading effects can occur. This raises a high demand for security assurance, with a high importance assigned to security evaluations.

This paper focuses on the description of our simulation environment for experiments aiming at evaluations of security of ICT systems underlying critical infrastructures. The description is proceeded with an outline of an approach for the characterisation and assessment of security threats to information systems of industrial critical infrastructures. The approach is based on the systematic planning, performance and analysis of experiments with simulations of attacks affecting control and supervision systems. It assumes thorough reconstruction of the information system of the evaluated critical infrastructure in a cybersecurity laboratory.

2. Critical Infrastructures Security Evaluation Approach

Our approach for evaluation of security of ICT systems of critical infrastructures is based on the simulation of attacks against the evaluated systems. To avoid *any* interferences with the systems, the experiments are performed in the secure and isolated setting of our cybersecurity laboratory. The approach comprises the following phases:

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

- *Reconstruction of the ICT system in the simulation environment.* Based on 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 which parts of the original system should be reflected completely and which subsystems can be approximated.

- *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 all in form of the descriptions of use scenarios.

The knowledge obtained in this phase will allow us to define the experiments where human factor plays a role (for example for the attacks requiring human interactions).

- *Design of experiments.* When designing the experiments, we first define the attack goals and the system sections which will be affected. Then we describe the attack scenarios depicting subsequent steps required for the successful attack. These textual descriptions we accompany with more formal attack specifications by means of attack trees. Finally we define the system conditions for the successful performance of the attacks and experiments (such as environmental settings, the required resources etc).

- *Performance of experiments.* Before each experiment we make sure that the simulation environment is in 'zero-state' – the initial state equal for all experiments. Then the image of the system settings is created in order to to make the experiments repeatable. After that we start performing the experiment. The system events are recorded.

- *Collection and analysis of results.* In the final phase we collect the information about the system events. We process it in order to extract the key, attack related, information. We analyse the information and formulate conclusions about the security of the ICT system. Where the systems exposes vulnerabilities, we propose countermeasures.

3. Simulation Environment

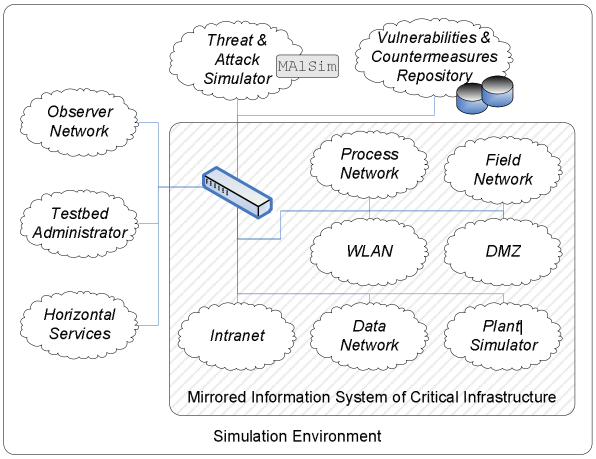

Figure 1. Simulation environment.

The simulations of attacks are performed in the simulation environment whose main part – *Mirrored Information System* (see Figure 1) aims at reconstructing the information system of the evaluated infrastructure. This part is flexibly configured depending on the particular needs. For example, for the infrastructure of a power plant we mirror the process network (interconnecting diverse subsystems of the energy production process), the field network (interconnecting controllers and field devices), the corporate network etc (see the figure).

Additionally the environment comprises the auxiliary parts which support configuration, performance and observation of the experiments; collection, storage and processing of results; and storage and provision of countermeasures: *Threat and Attack Simulator*, *Observer Network*, *Vulnerabilities and Countermeasures Repository*, *Testbed Master Administrator* and *Horizontal Services*.

3.1. Threat and Attack Simulator

Threat and Attack Simulator is the part of simulation environment where the simulated attacks are configured and launched.

There is a great number of already identified and documented attacks and there are also many attacks yet unknown to the computer security community. All of the attacks have diverse characteristics and they are performed in different ways. Some of them originate on one host, but usually they require a chain of hosts (to amplify the strength of attack or to make the attacker untraceable etc). The attacks require diverse attacker skills, from very basic (running an automated application) to very advanced (extensional developing skills, deep knowledge of computer system vulnerabilities etc). Furthermore to perform the attacks diverse types of applications and operating systems must be used. In general, attacks of different types can strongly differ in hardware (including attacker network topology) and software resources required to perform them.

Thus one of the primary characteristics of a system aiming at simulating attacks is its flexibility and easiness of configuration. The crucial feature of Threat and Attack Simulator is its flexibility in managing virtual subnetworks and creating multiple virtual network nodes. The network nodes, the hosts, are easily configurable and provided with diverse resources. Particularly they are provided with various software i.e. the operating systems (various Linux distributions, Microsoft Windows versions etc), and the specialised programs for developing attacker tools and for performing the attacks.

The topology of emulated fragments of Internet networks, as well as hosts which form a part of this topology, and their resources, change depending on needs. And the needs are determined by each particular attack. The subsystem is reconfigured depending on the simulated attack. The system topology is defined and designed in reference to each particular attack which is to be performed in its boundaries.

The attacks which are the most frequent in the Internet and which pose a serious threat against critical networked infrastructures, are based on malicious software (*malware*). To simulate the attacks we developed MAlSim – Mobile Agent Malware Simulator. MAlSim is a software framework which aims at simulation of various malicious software in computer network of an arbitrary information system. It is based on the technology of mobile agents, which appears to be particularly suitable for this application [4, 5].

3.2. Observer Terminal

The scope of the Observer system is to keep track of all the malicious or anomalous events happening in Mirrored Information System during tests and experiments.

The system is based on the Intrusion Detection Engine. The implemented architecture is composed of the following elements:

- Sensors, which capture the network traffic, analyse it and identify the traffic patterns which satisfy some predefined criteria.
- Observer Network, which delivers the packets selected by the sensors to a central collector.
- Observer Database, collecting data received by different sensors distributed in the whole system, and providing support for database querying.
- User Consoles, providing interfaces for access of data collected in the database.

For the implementation of the sensors we have adopted the Snort Technology [7]. The technology supports the sensor operation in four different modes:

- Sniffer mode – packets are captured and displayed directly on a console.
- Packet Logger mode – packets are logged to disk.
- Network Intrusion Detection System mode – the most complex and configurable setting, which supports analyses of network traffic based on matches against a user-defined sets of rules.
- Inline mode – packets are collected from `iptables` instead of `libpcap` allowing control over packet flow.

In our system, the sensors are set to the Network Intrusion Detection System mode.

The sensors track anomalies in the network and send alerts triggered by the discovery of such anomalies to the Observer Database. The alerts are first sent to the Observer Database server called Real-Time Repository. Then, after the experiment termination, they are automatically transferred to the second Observer Database server, namely the Archive Repository. This configuration allows keeping separated the data of past experiments and the data of the experiment in progress, which facilitates the analysis phase. Moreover keeping the two databases separated allows access to the data of past experiments without affecting performance of alerts collection of the ongoing experiment.

The graphical web interface for the remote access to Observer Database is implemented in BASE framework. BASE [1] is a web interface to perform analyses of intrusions detected by Snort. It provides authentication and access-control mechanisms.

3.3. Vulnerabilities and Countermeasures Repository

The aim of Vulnerabilities and Countermeasures Repository is to provide a structured knowledge base on vulnerabilities, threats, attacks of information systems and on the possible countermeasures. The system is composed of two sub-systems: the Vulnerabilities and Countermeasures Database and the Binaries Repository.

The Vulnerabilities and Countermeasures Database stores the knowledge about the existing and known vulnerabilities, threats, attacks and countermeasures. The Binary Repository is devoted to store and catalogue the 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.

We categorise the following types of information stored in the Vulnerabilities and Countermeasures Repository:

- Off-line documentation – documents of various type (text files, word processor documents, graphical files etc) stored locally in the file-system of the Vulnerabilities and Countermeasures Database.
- On-line documentation – information about vulnerabilities, attacks and countermeasures located in external web-repositories of the Internet. This information is linked to the database by external references.
- Binary codes – binary codes of patches, attack tools and other executables stored in the Binaries Repository.

Vulnerabilities and Countermeasures Repository was implemented in the framework of Industrial Security Risks Assessment Workbench (InSAW) [6, 3] which is our proprietary system based on a two tier client/server database driven architecture. Developed with Microsoft Visual Studio 2005 and Microsoft .Net Framework 2.0. InSAW facilitates assessment of vulnerabilities, threats, attacks and risks through a set of libraries and the graphical interface which allows creation of graphs and charts.

The system is composed of the following modules:

- SQLServer-based Database Management System for management of internal data repositories.
- Libraries, which store the information about Vulnerabilities and Countermeasures.
- Systems Repository – a collection of databases, one for each analysed system.
- Analysis Engines – analysis engines for the evaluation of different aspects related to vulnerabilities, attacks, threats and countermeasures.
- Querying, Reporting and Chart Units – facilitating data retrieval and representation (by means of reports, charts and tables).
- System Aided Modelling Unit, which provides the interface for modelling of vulnerabilities, attacks, threats and complex systems.

3.4. Testbed Master Administrator

The Testbed Master Administrator system brings in tools for remote control and monitoring of the hosts participating in the experiments. It manages the operations related to initiation and termination of experiments and allows real time observation of each system's behaviour when a simulation is launched.

Since the remotely controlled hosts are supplied with various operating systems (Windows and Linux), we employ diverse remote management solutions. For Windows we take advantage of UltraVNC and Windows Remote Desktop. For Linux, a remote management system based on SSH was applied, and the Putty SSH client was installed.

3.5. Horizontal Services

The Horizontal Services system is responsible for providing services that are needed for the efficient management of the simulation environment. At the current stage the two services have been implemented:

- Backup Service, for restoring initial simulation conditions before each experiment and to prevent from loss of data in case of hardware failure or accidental erasure. Backup Service is implemented using Symantec NetBackup software package, providing high-performance backups and restores for a variety of platforms, including Microsoft Windows and Unix-like operating systems.
- FTP Filesharing Service, supplying the simulation environment with a shared storage area which can be easily accessed from machines running diverse operating systems. The service is set up on open source FileZilla FTP Server, which supports FTP and FTP over SSL/TLS, with a secure encrypted connection between client and server. FileZilla also provides on-the-fly data compression significantly improving transfer rates.

4. Conclusions

In the paper we have presented our approach and the supporting simulation environment for experiments aiming at evaluations of security of ICT systems underlying critical infrastructures. The approach is based on the thorough analysis of the ICT system of the evaluated critical infrastructure and its reconstruction in our cybersecurity laboratory. In this configuration we implement attack scenarios to evaluate impact of the attacks, test robustness and identify countermeasures. The approach was successfully applied to the studies on security of a power plant [4], proving its operability, applicability and usefulness.

References

[1] Basic analysis and security engine. Internet, 2003. Available at `http://base.secureideas.net/index.php` (last access: October 30, 2007).

[2] E. Commission. Communication from the commission to the council and the european parliament: Critical infrastructure protection in the fight against terrorism. Internet, October 2004.

[3] I. N. Fovino, M. Masera, and A. Decian. Integration of cyber-attack within fault trees. In *17th European Safety and Reliability Conference (ESREL)*, volume 3, pages 2571–2578, June 2007.

[4] R. Leszczyna, I. N. Fovino, and M. Masera. Malsim – mobile agent malware simulator. Accepted for First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008), Marseille, France, 3 – 7 March 2008, March 2008.

[5] R. Leszczyna, I. N. Fovino, and M. Masera. Simulating malware with malsim. Accepted for 17th EICAR Annual Conference 2008, Laval, France, 6 – 8 May 2008, May 2008.

[6] M. Masera and I. N. Fovino. A service oriented approach to the assessment of infrastructure security. In *First Annual IFIP Working Group 11.10 International Conference on Critical Infrastructure Protection*, volume 1, March 2007.

[7] M. Roesch. Snort – lightweight intrusion detection for networks. Internet, 2003. Available at `http://www.snort.org/docs/lisapaper.txt` (last access: October 30, 2007).