Cogn Tech Work (2012) 14:143–155
DOI 10.1007/s10111-010-0171-2

ORIGINAL ARTICLE

The human role in tools for improving robustness and resilience of critical infrastructures

Aladino Amantini · Michał Choraś · Salvatore D'Antonio · Elyoenai Egozcue · Daniel Germanus · Reinhard Hutter

Received: 27 September 2010 / Accepted: 20 December 2010 / Published online: 19 January 2011

Abstract This paper presents a project dedicated to the development of means for improving the resilience of Critical Infrastructures (CIs) with respect to cyber attacks. The ability to sustain and protect the flow of information and data and the possibility to early detect, isolate and eliminate cyber hazards have become issues of paramount importance when developing the Supervisory Control And Data Acquisition (SCADA) systems of such a CI. The majority of tools dedicated to these goals are based on fully automatic autonomous self-reconfigurable systems that operate within the network, or online. However, the possibility to enable also human intervention for the further reduction in the vulnerability of CIs is equally possible. In this case, the intervention is considered offline and requires the active co-operation between a decision aid tool and a human operator. This paper presents a project aimed at improving robustness and resilience of CIs and discusses in particular the human interfaces associated with the offline tools. In essence, it is found that while the guidelines of the usability principle must be preserved, special account must be given to the type of issues involved and high professionalism of their users. This implies that certain basic criteria of the usability principle may be less relevant and their limitations may not be respected without loosing effectiveness and strength of the tools.

Keywords Critical infrastructures · Resiliance · Supervisory control and data acquisition systems · Cybersecurity

1 Introduction

Nowadays, security has become a major issue in improving people well-being, and it is usually associated with domains, such as public transport and civil aviation. However, another critical environment that requires particular attention is associated with the resilience of Critical Infrastructures (CI) that offer distributed services to a vast majority of the population, such as electricity or gas distribution systems, banking and communication networks and the like. These types of security issues are less prominent than, as an example, those in public transport because of the subtle and less evident impact that any malevolent

A. Amantini (✉)
KITE Solutions s. n. c., Via Labiena 93,
21014 Laveno Mombello (VA), Italy
e-mail: aladino.amantini@kitesolutions.it

M. Choraś
ITTI Ltd., Ul Rubież 46, Poznań, Poland
e-mail: michal.choras@itti.com.pl

S. D'Antonio
Consorzio Interuniversitario Nazionale per l'Informatica,
Dipartimento per le Tecnologie, Università degli Studi di Napoli
Parthenope, Centro Direzionale di Napoli Isola C/4,
80143 Naples, Italy
e-mail: saldanto@unina.it

E. Egozcue
S21sec Information Security Labs S.L, Pol. Empresarial La
Muga, 11, 31160 Orkoien, Navarra, Spain
e-mail: eegozcue@s21sec.com

D. Germanus
DEEDS Group, Technische Universität Darmstadt,
Hochschulstraße 10, 64283 Darmstadt, Germany
e-mail: germanus@cs.tu-darmstadt.de

R. Hutter
CESS Centre for European Security Strategies,
Grünwalder Str. 155a, 81545 München, Germany
e-mail: hutter@cess-net.eu

action has on the general public. Nonetheless, the effects of an activity affecting CI negatively can become devastating.

Many CIs rely on Supervisory Control And Data Acquisition (SCADA) systems, which are increasingly interconnected, and tend to make use of multi-purpose public networks. In particular, SCADA systems are the result of the integration of a large number of components, including ad hoc RTUs (Remote Terminal Units) and PLC (Programmable Logic Controllers), as well as COTS (Commercial Off-The-Shelf) components, legacy systems and Sensor Networks.

This leads to a landscape of heterogeneous data sources and communication technologies (wireless, ethernet, proprietary bus…) and makes SCADA systems susceptible to cyber threats and thereby put CIs and public safety at risk.

The vulnerabilities suffered by SCADA systems, and thus CI, may enable attacks directed against, at least, one of the following targets: (1) SCADA application, (2) operating system, (3) network, (4) field equipment, (5) typical IT applications.

To cope with the heterogeneity of data sources and in general with potential cyber attacks, a promising solution is the use of adaptable parsers. These are components in an interpreter or compiler, aimed at checking for correct syntax and at converting external event formats to an internal structure. In this way, parsers can elaborate the collected data. A novel approach can be defined and developed to SCADA systems security, based upon diagnosing as a key concept, associated with parsers. By diagnosis, it is meant the capability of (D'Antonio 2010):

(a) clearly identifying the causes of the attacks and
(b) accurately estimating their consequences on individual system components.

The approach to SCADA systems diagnosis consists in collecting information from several and diverse data feeds (network, sensors, RTU, SCADA server) using multiple security probes, which are deployed as a distributed architecture, and in performing sophisticated analysis of intrusion/fault symptoms. This makes it possible to escalate from intrusion/fault *symptoms* to the actual *cause* of the intrusion/fault, and to assess the *damage* in individual system components. Recovery actions are then performed in order to eliminate or mitigate the effects of the intrusion/fault.

Following this idea, a number of tools have been developed and their performance demonstrated for the improvement of communications and control processes in real-time operation (online). At the same time, another set of tools has been developed, for the purpose of better treatment and assessment of vulnerabilities, for the preparation and development of security measures (offline).

These tools are totally transparent to the user in that self-configurable architectures can be designed that enable:

a. to detect in a short time a "node" under attack or susceptible to suffer an attack and take adequate countermeasures in order to protect the CI, as well as
b. to develop of means to fight and resolve the situation with no effects on the final user of the infrastructure services.

This whole process seems to indicate that a fully automated system can be devised.

However, at certain levels, a human being is called into play for critical decision-making, especially when the vulnerabilities of certain networks are found and there is need for selection in order to implement the plans as recommended by the system of intervention. The level of decisions involved in the process of exploitation of inherent resilience of these novel systems requires complex and articulated knowledge by the user, comprising a good understanding and vision of the CI architecture, of the potentiality of the new type of SCADA systems and of the resilience philosophy that guided the design of the CI.

This complexity and the inherent need of rapid intervention imply that the interfaces of these support tools have to be extremely efficient and effective. Consequently, an essential process of interface design has been developed that follows the basic principles and requirements of the User-Centred Design and Usability (Bevan and Macleod 1994; ISO 1993, 1999), which are established and proven means for ensuring the best exploitation of IT means utilized by human beings.

In this paper, an overall view of the integrated system will be given, by presenting the objectives, scope and state of work of the development of the overall technology for improving the resilience of CIs. Then, the implementation of appropriate interfaces for the offline components will be discussed in combination with the issues of usability and effectiveness. The sample cases that are planned in order to demonstrate the validity of the overall approach and of the associated decision support system will be presented. Finally, the conclusions will give preliminary perception of possible future implementations of the proposed methodology and instruments in real complex and critical infrastructures.

2 Overall description of the proposed methodology

The key issue of CIs and their monitoring and control systems (the SCADA systems) is the ability to manage the landscape of heterogeneous data sources utilized in the integrated systems. SCADA can be described as centralized systems that monitor and control entire sites, or

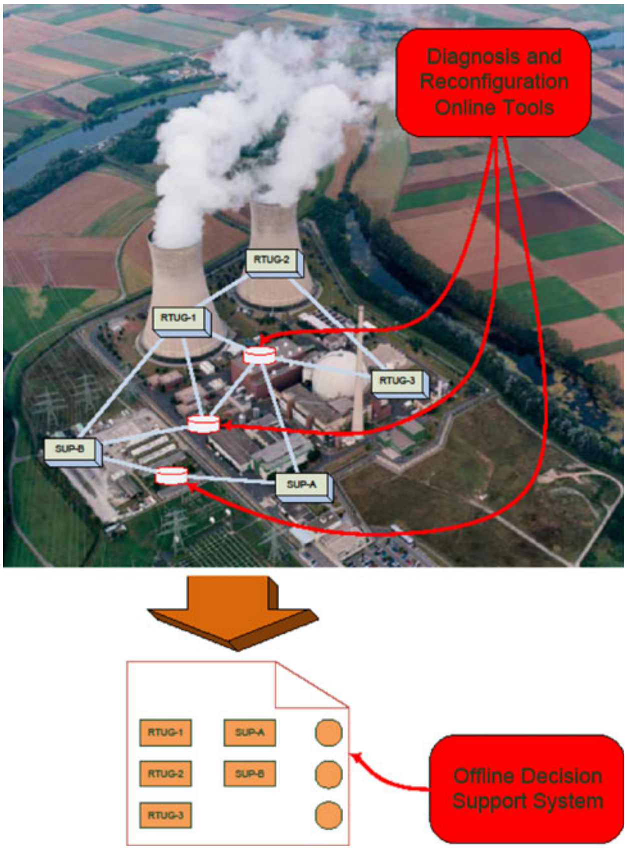

Diagnosis and Reconfiguration Online Tools
RTUG-2
RTUG-1
RTUG-3
SUP-B
SUP-A
RTUG-1
SUP-A
RTUG-2
SUP-B
RTUG-3
Offline Decision Support System

Fig. 1 A Nuclear Power Plant (CI) and its associated SCADA system

complexes of systems spread out over large areas, anything between an industrial plant and a country. The latter can be identified as Critical Infrastructures (CI). As an example of a CI and associated SCADA system see Fig. 1.

In order to improve resilience of the CI, a number of steps, or phases of intervention, need to be considered. Primarily, it is possible to distinguish between autonomous self-reconfigurable systems that operate "in-network", or online, without the intervention of an external human operator, and a second type of intervention, called "off-network", or offline, that requires the active co-operation between a decision aid tool and a human operator. While this paper concentrates on the latter means of intervention, in order to present a more comprehensive view of the project, the first type of solution is also briefly described hereafter.

2.1 The online tools

The online type of intervention focuses on: analysing and modelling dependencies between critical infrastructures and underlying communication network; designing and implementing traffic engineering algorithms to provide SCADA traffic with quantitative guarantees; exploiting Peer-to-Peer (P2P) overlay routing and replication mechanisms for improving the resilience of SCADA systems, or defining a self-reconfigurable architecture for SCADA systems; and developing diagnosis and recovery techniques for SCADA systems.

These different tools can be summarized in the *diagnoser,* the *MPLS reconfigurator* (Multi Protocol Label Switching) and the *P2P-based middleware*.

2.1.1 The diagnoser

In a first phase of intervention on SCADA systems, adaptable parsers can be utilized that can convert external event formats into an internal flexible structure enabling following elaboration of collected data. Depending on specific implementation choices and/or deployment requirements, such components may be implemented/deployed as relatively autonomous software modules or integrated in specific sub-systems of the diagnostic framework.

Adaptable parsers gather information from multiple data sources distributed across the SCADA system. To cope with the heterogeneity of the formats of such data, their highly structured organization can be exploited by adopting grammar-based parsers. Adaptable parsers translate raw events into an intermediate format, so that they can be merged in a single data stream for further processing. The outputs of this phase are provided to diagnosers.

The diagnostic framework is able to handle multiple classes of faults/attacks to a SCADA system, e.g., hardware-induced software errors in the application, process and/or host crashes or hangs, performance degradation due to intrusions, and more. Three levels of interventions are envisaged:

1. First, events are screened, and the main stream of data and events is split in into a set of individual streams consisting of events, which share a common semantic. Data irrelevant for diagnostic activities are filtered out.
2. Second, a further refinement of the streams is operated, and possibly, adjustments on events are operated.
3. Third, low-level events, i.e., events related to specific aspects of individual components of a SCADA system, are aggregated and translated to high-level events, i.e., events relevant in a resiliency-oriented view of the overall SCADA system. The transition from low-level events to high-level events is made possible by the use of ontologies.

Ontologies represent an effective tool to describe complex and interdependent symptoms related to faults and attacks, since they are meant to provide formal specification of concepts and their interrelationships. An ontology-based hierarchical organization of event patterns is used to automate the process of deriving queries for diagnostic analysis. The knowledge expressed in the ontology allows the diagnostic process to identify which faults/attacks can be hypothesized to be the cause of identified symptoms.

The processing must be performed in a timely fashion. Valuable tools allowing near real-time elaborations over large volume of data come from the research field referred to as "stream processing". A Complex Event Processor engine looks for patterns of events in the data streams being processed.

The results of the diagnostic procedure are then used to select the most appropriate and efficient reconfiguration action to be performed. Two reconfiguration technologies can be considered for implementation:

(a) Multi-Protocol Label Switching (MPLS)–based traffic engineering algorithm for resource optimization and network resilience; and
(b) Peer-to-Peer (P2P) overlay network for path and data redundancy.

These two "reconfigurators" provide efficient bandwidth usage, data replication and data access during perturbations. The complementary of the reconfigurators contributes to reconfigure the system and mitigate the effects of intrusions or faults.

2.1.2 The MPLS reconfigurator

Once a network node in the SCADA system is found to be under attack or compromised, it is necessary to react promptly. The fastest countermeasure that can be put in place is to avoid that packets continue being forwarded to the attacked node. In this way, the information carried by those packets is not exposed, and the system administrators have all the necessary time to take the appropriate actions to repair the node.

The techniques used to route packets around the affected node vary depending on how routing is performed in the normal network operation.

The solution proposed is based on the computation of a pair of disjoint paths between every two network nodes that act as source and/or sink of information messages.

One solution is to make one of the disjoint paths, the active path, and keep the other one as a backup path to be used in case a node along the active path fails or is found to be attacked/compromised. In such a case, recovering from problems affecting the normal operation of a node along an active path would just require the activation of the backup path.

Another option is to simultaneously use both the active and the backup path by "splitting" the packets of a flow along the two paths. This technique can be used to prevent information eavesdropping before the attack is actually detected. Indeed, each node along either the active or the backup path will not receive all the packets of a flow, and thus, an attacker that has compromised a single node will not be able to reconstruct the whole information flow.

2.1.3 The P2P-based middleware

The Peer-to-Peer (P2P)–based middleware has the goal to increase the SCADA system's resilience during perturbations.

The middleware is installed on Remote Terminal Units (RTUs) and Master Terminal Units (MTUs) in the SCADA system and is subject to a middleware design. A node-level architecture of the P2P middleware is presented in Fig. 2. SCADA messages are intercepted, forwarded and processed. The processing in the middleware layer includes extraction of SCADA message payload that requires a SCADA model that describes the payload structure of the SCADA protocol's messages. Furthermore, processing involves SCADA payload storage in a distributed hash table (DHT) of the P2P overlay network. Benefits from P2P technology are path redundancy and data replication. These benefits are taken advantage of to counteract perturbations like node crashes or data corruptions and thereby to achieve an increased system resilience and robustness.

A schematic overview of P2P-enhanced SCADA communication is given in Fig. 3. Different P2P protocols have been evaluated in simulations (Germanus et al. 2010), and P2P protocol decision criteria for different system classes have been developed (Khelil et al. 2010). The test-bed implementation of the P2P reconfigurator is based upon the Kademlia protocol (Maymounkov and Mazières 2002) since it has shown good results in simulation runs. The P2P network is closed, i.e., admittance is required to join the network. Therefore, common attack vectors against P2P systems like the Sybil attack (Douceur 2002) are prevented.

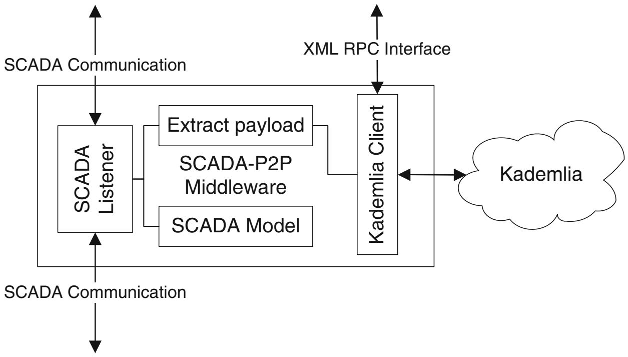


Fig. 2 Node-level architecture of the P2P-based communication middleware

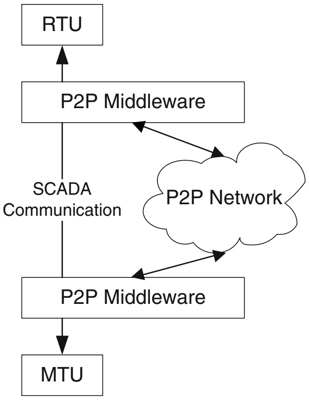


Fig. 3 Network-level architecture of the P2P-based communication middleware

2.2 The offline tools

The offline tools are dedicated to the generation of the necessary information for the user (*operator*) who is the pivot element in an intervention in the case of attach on a CI.

The major components of the Offline Security Assessment Framework are (Fig. 4): the Vulnerability Collector (VC), the Ontology Handler (OH) and the Decision Aid Tool (DAT).

2.2.1 The vulnerability collector

Vulnerability collector is an aggregator and indexing tool used by the Security Assessment Framework. The main objective of this component is to automatically aggregate vulnerabilities, weaknesses and threats from different external sources and consolidate the information in a single centralized place.

The Vulnerability Collector offers different kinds of interfaces (human and programmatically handled). For example, it offers the Ontology Handler the possibility to perform Request and Registration for updates operations as a means to make use of the information collected in the repository.

2.2.2 The ontology handler

The Ontology Handler is in charge of providing the Decision Aid Tool with knowledge relative to SCADA systems' infrastructure and security information. This knowledge is stored within a specific Security Ontology (Choras et al. 2009; Kozik et al. 2010). The services, operating systems, equipments and networks composing any SCADA system are discovered by the Ontology Handler aided by the discovery tool module. They are all considered assets. In addition, security information is retrieved from the Vulnerability Collector enabling to associate Vulnerabilities, Threats and Safeguards to the corresponding affected assets.

2.2.3 The decision aid tool

INSPIRE Decision Aid Tool (DAT) is a decision support system basing on the knowledge stored in the Security Ontology. Therefore, this ontology provides knowledge about SCADA components, their vulnerabilities, related threats with risk level and some countermeasures.

DAT is able to read ontology classes and instances, and reason about them to provide some countermeasures, advices for user of DAT. DAT uses ontology classes and instances to get the knowledge in the RDF triples format

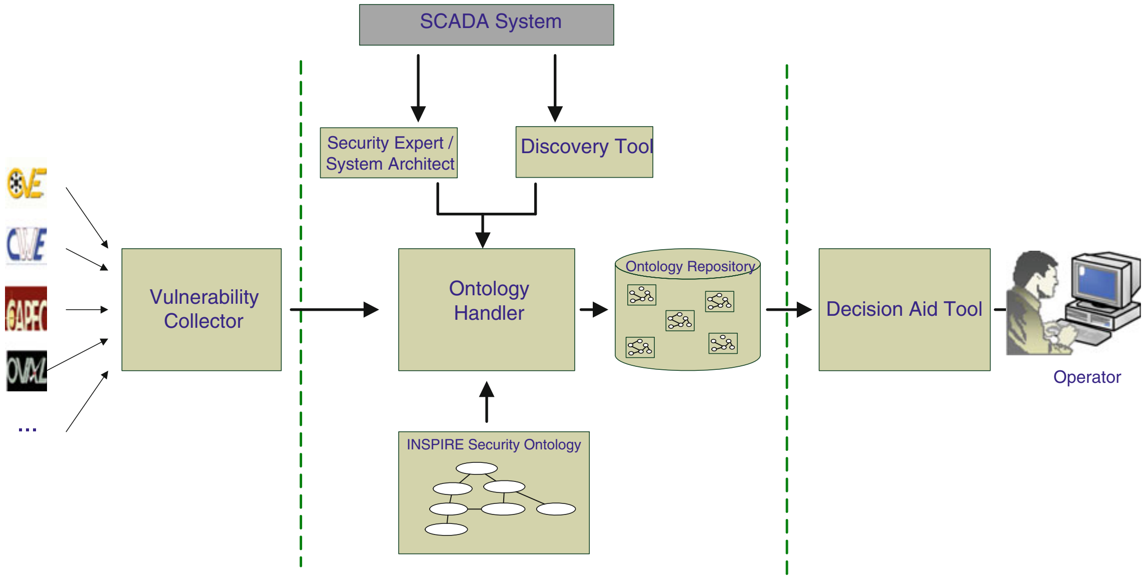


Fig. 4 Security framework architecture overview

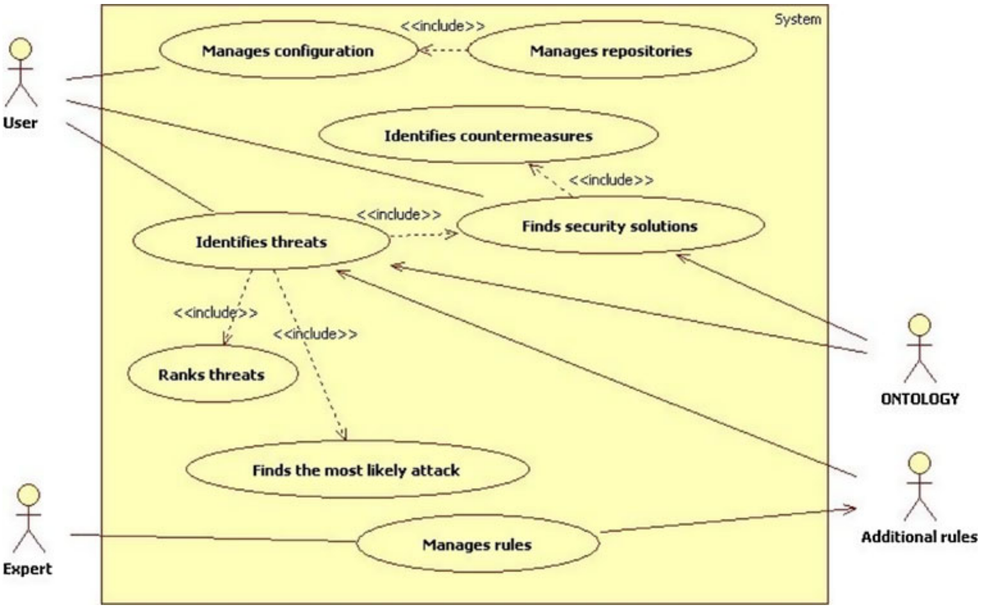


Fig. 5 DAT functionalities

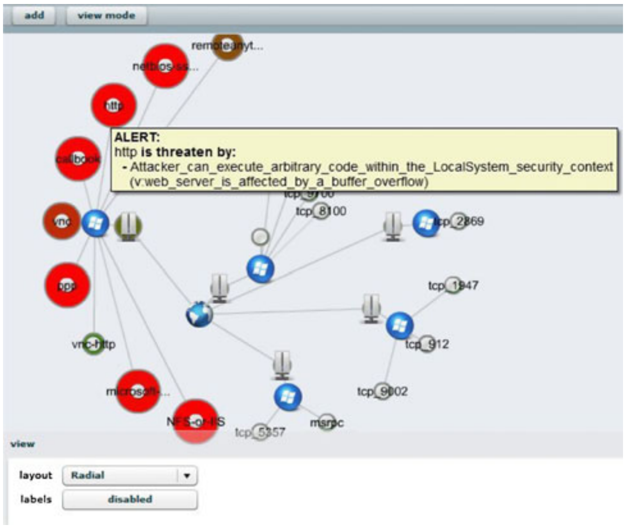
and processes them in rule engine. RDF triple is represented by subject, predicate—property and value.

The main feature of DAT is ranking threats related to particular vulnerability from the level of risk point of view. Other DAT functionalities are presented in UML diagram in Fig. 5.

Threat analysis performed by DAT is divided into three steps. First, the topology diagram is rendered to provide the operator with the information, e.g., about elements interconnections, used applications, etc. In the next step, the threats are visualized by the red blinking nodes (Fig. 6).

At the end, a security report is presented. The sample result of CI system security evaluation by DAT is presented in Fig. 7. The presented security report contains the ranked threats for discovered assets with their threat severity value calculated by a Bayesian network.

More details about combining ontology inference engine with the Bayesian network in order to calculate threat severity risk value are given in Choras et al. (2009) and Kozik et al. (2010). Moreover, in the security report, details about the detected threats and the proposed solutions are given.



ALERT:
http **is threaten by:**
- Attacker_can_execute_arbitrary_code_within_the_LocalSystem_security_context (v:web_server_is_affected_by_a_buffer_overflow)

layout: Radial
labels: disabled

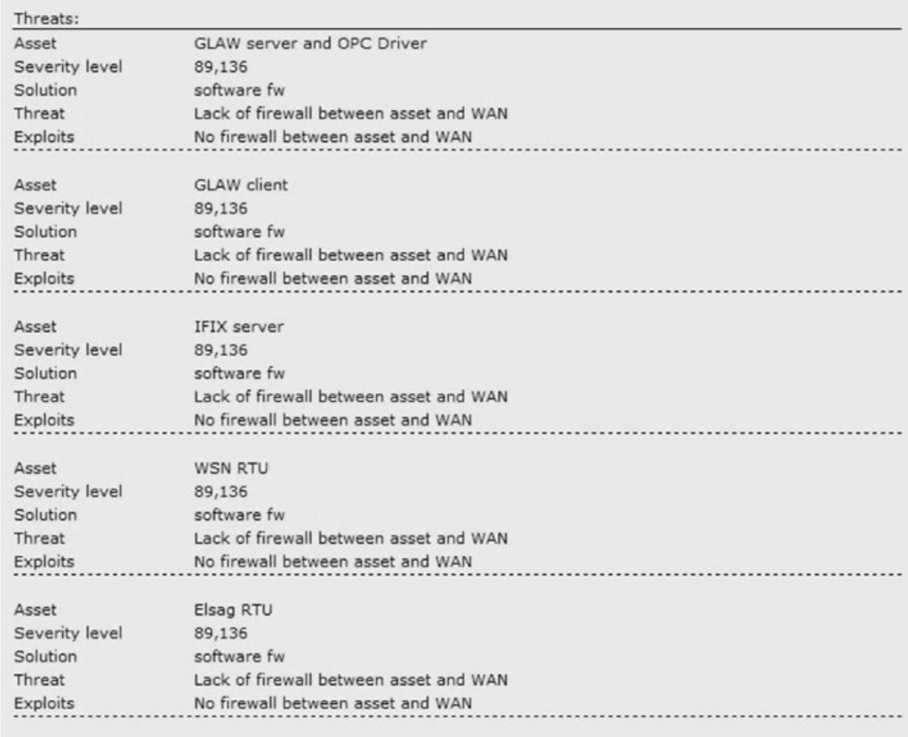
Fig. 6 Topology diagram and visualized threatened nodes

3 Actors and interfaces of the offline tool

3.1 Classical approach to user-centred design process

The design of support systems requires an approach, which effectively reflects user needs. The development of the *user interface* necessitates an accurate design approach, in order to guarantee that the objectives of the support tool are reached effectively and efficiently, without disturbing, annoying or event increasing the workload of the user. The implementation of an iterative process of interaction between designers of instruments and users has been nowadays widely accepted as the most appropriate ways to develop useful and effective instruments (Cacciabue and Martinetto 2006).

These aspects are particularly important in domains where an enormous variety of users exists, in terms of age and experience, with little levels of formal training and experienced in different "standards" of user interfaces. For these reasons, the design of a support system aimed at improving the robustness and resilience of Critical Infrastructures requires consideration of a well-prepared usability concept. Two fundamental principles for designing Human Machine Interfaces (HMI) must be accounted



```
Threats:
Asset               GLAW server and OPC Driver
Severity level      89,136
Solution            software fw
Threat              Lack of firewall between asset and WAN
Exploits            No firewall between asset and WAN
------------------------------------------------------------

Asset               GLAW client
Severity level      89,136
Solution            software fw
Threat              Lack of firewall between asset and WAN
Exploits            No firewall between asset and WAN
------------------------------------------------------------

Asset               IFIX server
Severity level      89,136
Solution            software fw
Threat              Lack of firewall between asset and WAN
Exploits            No firewall between asset and WAN
------------------------------------------------------------

Asset               WSN RTU
Severity level      89,136
Solution            software fw
Threat              Lack of firewall between asset and WAN
Exploits            No firewall between asset and WAN
------------------------------------------------------------

Asset               Elsag RTU
Severity level      89,136
Solution            software fw
Threat              Lack of firewall between asset and WAN
Exploits            No firewall between asset and WAN
------------------------------------------------------------
```

Fig. 7 Security report with the ranked threats

for, namely, the strong regard of System Usability criteria, and a User-Centred Design process (UCD).

The International Standard Organisation has defined System Usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO 1993, 1999). The definitions of effectiveness, efficiency and user satisfaction are also proposed by ISO. However, they do not represent a precise instrument to measure the usability level of a system, as they do not exactly define what features make an object usable. There are no general universal usability rules. Usability is a relative concept depending on three factors, namely task, user and environment (Bevan and Macleod 1994). These three factors combine in generating the particular context of system use.

As the main purpose to support the security and the protection of SCADA networks is a very complex goal, several aspects and elements have to be considered. The wide range of threats to which networks are nowadays exposed needs dedicated tools and strategies to face critical information protection.

3.2 The users of the offline tool

To understand the perspective adopted in the design phase of the interface, two different types of users have been considered: the *security expert* and the *tool administrator*. Each of them has specific duties and tasks to perform and is foreseen and supported by specific interfaces.

3.2.1 The security experts

The *security expert* of the offline tool should be a figure having long experience and deep knowledge in the security domain and in the company systems. This user needs to have a thorough understanding of the control system architecture. It should be aware of major information regarding end-systems: the set of machines (e.g. SCADA servers, RTUs, etc.) comprising the system, the operating systems in use or the running applications on each machine. The *security expert* should also be aware of the communication systems in place, the network segments (e.g., VLANs) defined, network points of access, system providers and maintainers, and/or business-related connections to corporate networks. Finally, this figure should be involved in the definition of the security policies affecting the control system or at least be aware of them. For instance, the communication requirements among network segments and how they should be satisfied, or the way in which new patches and workarounds should be managed to get rid of critical vulnerabilities, or the way to access control systems from the outside for maintenance purposes from the SCADA vendor.

The *security expert* user may have two different roles when using the tool: security operator and security advisor:

- As a security operator, the user would be in charge of using the offline system to analyse the current configuration of the SCADA network to discover misconfigurations, hazards and any possible source of impairment of the good level of performances. When notified of inappropriate network, software or hardware configuration, the security operator has to apply countermeasures, possibly according to the offline tool suggestion included in the security report and company's security policies.
- As a security advisor, the user would help enriching the tool and adapting it to the specific scenarios, thanks to its expert knowledge. Some of these tasks are: consolidate and disambiguate the information data retrieved from external sources, decide whether a new vulnerability repository could be incorporated into the system, define new expert rules for the DAT, or complete and enrich the target system's ontology with new assets, services, applications or vulnerabilities.

3.2.2 The tool administrator

The *tool administrator* should have a thorough understanding of the offline tool. He/she should have a deep understanding of how the tool works and how it is configured and should support the *security expert* when acting as a *security advisor*.

For instance, the *security expert* user would define new expert rules for the offline tool. Then the *tool administrator* would incorporate these rules to the offline tool, removing obsolete ones and adding the new ones to the precise rules' path. The *tool administrator* would also be responsible for adding new security repositories into the offline tool, and configuring the crawling and scheduling options for these new repositories based on the *security expert* advice. Moreover, since the operation of data collection is an automated process that relies on different sources of information, ambiguous or not fully coherent data can be gathered. In this case, the intervention of the *tool administrator* is required to preserve the correctness and consistency of the data.

3.3 The interfaces of the offline tools

The three components of the offline security framework, namely the Vulnerability Collector (VC), the Ontology Handler (OH) and the Decision Aid Tool (DAT), provide their own administration interface, each with its own specific functionalities. For operating the OSF, a unique interface is provided by means of the DAT Operator interface.

3.3.1 The VC administration interface

The VC administration interface is a web-based GUI (Graphical User Interface) that allows the component maintainer to perform the required operations to configure the system, and all the information regarding the main application bootstrap. This facility includes the creation of users, access keys, e.g., licenses for the web service, and permission management. This GUI also allows the tool maintainer to configure the batching and crawling processes, manage the existing sources, the data field mapping and control the batch scheduling options. Since a security expert would be willing to enrich the security information database built up with information from external security sources, the VC administration interface also allows creating, modifying or deleting existing information in the database.

As mentioned above, a consolidation and disambiguation process might be sometimes necessary. This is done using the VC's administration. A queue with unresolved issues is presented to the user for quick accept/decline/merge operations over the recently introduced changes.

Finally, database information backup and restore operations could be available through this GUI. This allows to readily create a powerful GUI for prototype systems without investing time and resources on unnecessary heavy development tasks. Everything in the VC makes use of the web-based features. The use of rich html descriptions allows the inclusion of images, bolded and Italic texts, and other HTML-based items.

3.3.2 The OH administration interface

In the case of the OH, the administration interface provides a graphical interface to users, who are generally, or at least should be advised by, experts on SCADA systems, enabling them to launch a network scan and to store the discovered assets within an ontology instance of the ontology repository.

This GUI also enables users to visualize those ontologies that are in the ontology repository. Users have the possibility to load them and to enrich them by adding missing assets or by renaming discovered assets. The administration interface comprises four different panels, each with its own functionalities. The top panel contains a set of buttons used to launch the discovery process, to set the name of the newly created ontology and to retrieve security information from the Vulnerability Collector. The left panel is used to display the list of all the ontology instances stored by the Ontology Repository component. The central panel displays all the assets contained in an ontology instance. Finally, the right panel contains a toolbox with buttons to add elements that have not been discovered and/or enrich the information automatically gathered. OH's administration interface has been implemented in Java. JavaFX was used for the GUI. This is a new software platform that is built on Java technology for creating and delivering rich Internet applications. Furthermore, Jgraph, an open source Swing graph component, is used to visualize and manipulate graphs.

3.3.3 The DAT administration interface

The DAT administration interface is a GUI intended for the tool administrator and advised by the security expert. It allows to maintain security rules, i.e., facts about system and relations between its elements, and therefore, it helps in enriching the knowledge stored in ontology provided by the OH component.

In a prototype version of the offline security system, this GUI is not separated from the operator interface. However, for a commercial product, the three maintenance interfaces (VC, OH and DAT) should be merged into a single one and separated from the operating interface.

3.3.4 The DAT operator interface

The System Operator interface, which is provided by the DAT, is a user-friendly graphical interface that allows to identify and rank threats found in critical infrastructures. Furthermore, the DAT allows users to find security solutions for a particular threat and finds adequate countermeasures and strategy for minimizing the risk. What is more, DAT allows the *security expert* operator to perform simulation scenario, answering the questions such as "what may happen if particular action is taken" or "what happens (in terms of security) when particular equipment or software is added".

This GUI clearly defines the number of functions necessary for a correct operation of the system. These functions are grouped into several menus. The main menu is the Start DAT, which provides a set of different functions to interact with the system, namely a function that provides a mean to load a new ontology instance, a function that aims at finding threats and areas vulnerabilities prone to be exploited by a specific threat, and a function that displays a sorted threat rank, based on severity level. This includes a list of the affected assets, threats and the severity score. It also provides a detailed overview of each pair of asset-threat, which helps the security expert to get a comprehensive understanding of the real state of the control system in terms of security.

3.4 Critical issues of the interfaces for real applications

As described previously, offline tools interfaces are web-based applications. In addition to the undeniable advantages implied by this choice, some remarks can be made.

3.4.1 The browser

The front end of a web application is the browser of the user; however, different browsers are available each one with its own characteristics, features and bugs. This implies a dependency on client products, which may affect the web application behaviour. In fact, they can introduce bugs, compromising the correct execution of programs, or differences in the rendering of HTML elements, altering the look-and-feel of the application. The chosen software and even its version can potentially alter the correct behaviour of the user interface.

Web browsers are software applications that a user needs to operate with a web server. Web browsers request information or resources to web servers, which in turn locate these resources and send them back to the web browsers as a web page (HTML document), XML document, etc. There are different web-based applications and programs created with different programming languages (Active X, Java) and scripting languages (Perl, Java scripting, PHP) and AJAX (Asynchronous Javascript and XML). In any case, the browser has to display the requested information on a computer screen in an appropriate manner.

Web browsers are made up of two components, a user interface and a rendering engine. The most known engines are Webkit (Safari, Chrome), Gecko (Firefox, Flock, Camino), Trident or MSHTML (Internet Explorer), Presto (Opera), KHTML (Konqueror). The rendering engine depends on the type of operating system and browser. Therefore, this represents a problem for web application designers, as an application could work in a web browser but not in others.

The security of web browsers is very important in IT security. A vulnerable web browser and lazy security updating procedures may attract a variety of threats like spy-ware, attackers which could take control of the computer running the browser, or the like. Actually, attacks that take control of browser vulnerabilities are increasing, so it is significant to choose browsers that address application needs not introducing too many vulnerabilities. The storage of private (sometimes unauthorized) information in web browsers, e.g., authentication credentials, is another threat that could pose new problems. One needs to keep in mind that the focus of attackers is to take control of computers, steal information, destroy files or use this kidnapped computers to launch coordinated attacks to specific computer system targets.

There are many scripting technologies, such as ActiveX, Java and Javascript or VBScript. These technologies give extra functionality to traditional web browsers, but many of them also introduce more severe vulnerabilities. A poor implementation, an insecure configuration or a bad design could be a reason to make web browsers more vulnerable. Some web browsers have the option to disable these technologies, while others may permit enabling features on a per-site basis. Nowadays, it is very usual to install more than one browser on the same computer, so it is important to configure securely every web browser that may be installed on the computer, because people manage very sensitive information. If there is more than one browser available, it is recommended to use one browser only for important or sensitive transactions and the other browsers for common use.

Threats affecting browsers are (among others) buffer overflows in utility technologies (Flash, ActiveX), permission and access vulnerabilities (browser plugins accessing the hard drive files) and cross-site request forgery requests (location of page, url or resource is false and can be forged by rogue web applications).

3.4.2 Application back end

Another important issue is the security and usability of web applications back end. There are different parts that may comprise of back-end systems, such as databases, complete content management systems (Drupal, Joomla), HTTP servers, application technology servers (Java—tomcat, php) and others that interact with the user at some point. All these technologies compound the platform, and the platform may be affected by vulnerabilities in any of the components. Moreover, all the components may affect each other's security status and trustability. It is important to design and configure the complete platform to avoid cross-technology attacks and weaknesses. The browser interacts with the application, which is provided by the application technology server, supported by the application server and using information from a knowledge database, using an HTTP server to communicate both channels (client and server). The connection between all these technologies could introduce (and also mitigate) some vulnerabilities that may only happen when all the components are put together.

Security in the whole framework is important because the result is also interpreted by the user's browser applications. It is not only that the platform can be compromised using a vulnerability, but it can also be used to attack application users through their browser applications. Threats affecting the back end of an application are very heterogeneous, going from parameter tampering: including Database language injection (interpreted by the database engine), Application technology injection—php or java injection—(interpreted by the application technology server), Application server issues (including permissions, default configurations) and HTTP server vulnerabilities.

As stated before, security is always a major concern in web- and http-based application and user interfaces. Many well-known threats may affect applications using these technologies: XSS, SQL-Injection, directory traversal, spyware, etc.

3.4.3 Secure access to security information

Security information contained in the VC is valuable for the tool to work as expected. Therefore, it should be somehow protected. For these reasons, the VC includes a Roled-Based Access Control (RBAC) system, which allows handling profiles and users for authentication and authorization processes. Those applications being connected to the exposed web Services (using SOAP) are treated as a regular user (thus they are authenticating using an API key linked to a user account). This way, the role-based access controls applied in the VC, can easily be extended to the Ontology Handler based clients, as well as to other applications making use of its web services API.

The access control applied currently in the VC includes single authentication factor (password-based credentials) and role access security differentiation between not-logged-in-users and logged-in-users. The *tool administrator* user introduced in the previous section is further refined in the VC component. There are special features that can be assigned to the tool administrator with full administration rights or to the tool administrator with a more restrictive permission set (e.g., maintainer technician) to operate with the regular processes of the VC: adding new/removing feeds, disambiguation or any other function not related to the application configuration. For example, a basic maintainer technician may be able to create users if the permissions are assigned to that role, which is a feature reserved by default to the administrator role of the VC.

Role-based permissions go far beyond the access/deny rule, including extended granularity for feed management (access, view, edit own, edit any, delete own, delete any) but also includes user commodities to enhance application usability (permission to change language of the application for its own account, change time zone, etc.) that allows a nicer and smooth navigation and usage of the application.

The modular nature of the selected CMS allows additional identification factors to be applied without any modification by enabling the required functionality. Examples could be the usage of certificates for user identification, external authenticators (like openid or other services) and additional key factors (tokens or similar elements). Furthermore, session security is already covered by automated logout, inactivity management and session restriction by IP address or number of sessions.

General security measures have also been applied in the whole application regardless of the operator usage. In the case of recovery and error handling, a backup of the application is performed regularly as an autonomous process, with the interface for the site administration to perform backup and recovery operations on demand.

Everything in the interface could be provided in the user language (assuming that the content has already been translated to that language) as the application supports i18n extensions and localization functionalities, including timestamps and date/time format according to each language (Fig. 8).

4 Sample cases under development

A number of demonstrations and test cases have been devised with the purpose to show the capabilities of the tools discussed above, i.e., both online and offline tools, for all phases of the security provision process. These phases, which are covered with different intensity and level of detail, can be identified in:

- Prevention/mitigation phase, e.g., hazard assessment and risk analysis performance.
- Preparedness phase, e.g., contingency planning.
- Response phase, e.g., definition of countermeasures in cases of exploitation of system vulnerabilities by attackers.
- Recovery phase, e.g., application of recovery techniques for SCADA systems.

While most of these tests are associated with online tools and thus do not imply an intervention of an external human operator, the evaluation of the Decision Aid Tool (DAT) and its interface are part of the offline system sample case. This will be briefly discussed hereafter.

The goal of the sample DAT scenario is to show “simulation mode” of the DAT, which allows to evaluate the risk of particular action prior to the physical manipulation. The demo uses ontology provided by “Topology Discovering Tool” and “Expert knowledge” provided by expert via DAT GUI interface (Fig. 4). The user, in interaction with the DAT, carries out the following steps:

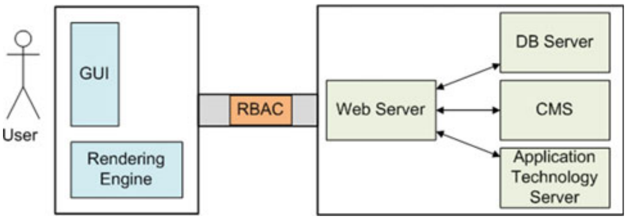


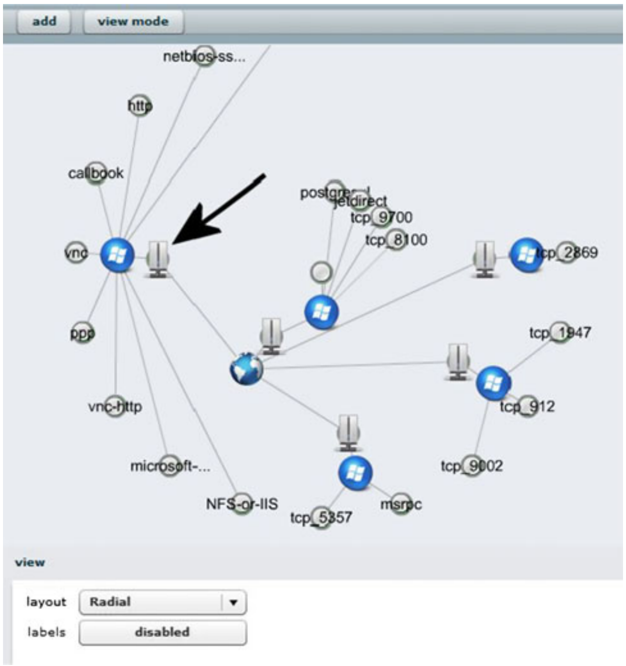
Fig. 8 Critical issues of application interface

- Exploitation of DAT to visualize topology graph;
- Turn-off of firewall and ant-virus applications on one of the routers;
- Identification of the hazards and risks and visualization of topology again; and
- Simulation of installation of firewall and anti-virus application on affected network.

At the beginning of the demo, the system topology is analysed. An example of topology of test bed is shown in Fig. 9. The arrow indicates the router where the firewall and anti-virus software will be switched off.

As it is shown also in Fig. 10, a single action has impact on many nodes being in relation with affected router. The cascading effect can be noticed. Turned-off firewall and anti-virus protection exposes the Windows 2000 operating systems to different network attacks, which eventually has impact on the provisioning of the applications hosted by that Operating System. What is more it is also assumed that the IIS WWW server with WebDav service is enabled on Windows 2000 (default configuration is assumed due to the lack of detailed information about ran applications and services).

The majority of discovered problems can be solved by installing anti-virus and firewall applications on Windows



layout Radial ▾

labels disabled

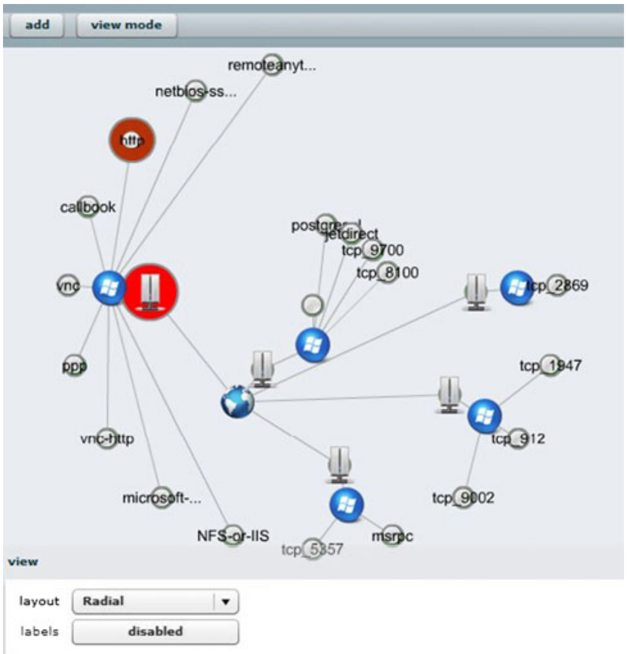
Fig. 9 Topology diagram before security level assessment

2000. Therefore, the user inserts into DAT the information that such an action has been taken and as result new security report is obtained (the visualized topology can be shown in Fig. 10).

5 Comments and conclusions

In this paper, a novel approach for the improvement of the resilience of CIs with respect to cyber attacks. Two methods of intervention have been considered. One based on fully autonomous self-reconfigurable systems that operate online, without the intervention of an external human operator, and a second one, or offline, that requires the active co-operation between a decision aid tool and a human operator. This paper has analysed the problems associated with the latter means of intervention. In particular, the implementation of appropriate interfaces for the offline components has been studied in combination with the issues of usability and effectiveness.

On the one side, it has been ascertained that the iterative process of interaction between designers of tools and associated interfaces and users must be preserved in order to reach the most effective results of the means offered to the user. On the other side, the different types of users and the specific expertise required for the use of the tools require particular consideration. In this case, the two main types of users, namely the cyber *security expert* and the *tool administrator*, are both specialists in Information Technology and have deep knowledge of in the management of networks and data exchange. Therefore, they are familiar and extremely confident with GUI containing many different and diverse information as well as with exploiting multi-paging and multi-criteria systems. Therefore, certain criteria of friendliness and usability, such as the complexity and amount of information displayed on a screen or the need for the user to find needed data and information on different pages and screen displays, are not, in this case, crucial problems.



layout Radial

labels disabled

Fig. 10 Topology diagram and visualized threatened nodes after installing the firewall and anti-virus applications

An issue that is of importance in the development of the interfaces is the fact that particular attention needs to be dedicated to the overall goal of the tools underdevelopment. Indeed, also in the case of the interfaces, the different types of software, data management and application technology servers can offer vulnerabilities and must be adequately selected, protected and controlled.

These two aspects lead to the consideration that the designer of an offline tool can be relatively free to generate complex and articulated interfaces relaying on the ability and expertise of the users. At the same time, the selection of the IT means and architectures has to be very carefully assessed before implementation.

This approach, however, does not avoid the implementation of the iterative process of assessment of the usability of the interfaces. In fact, once the designer has generated an interface with its information and means of actuation, even though complex and articulated, it is necessary to evaluate the way in which an expert user will make use of it and will exploit the various options and alternatives.

Consequently, on the one hand, it will be possible to develop complicated and articulated GUIs, as the users are experts and knowledgeable in the management of IT instruments. On the other side, the assessment of the most effective usage of the tools and the complete synchronization between designer and user can only be achieved through a detailed set of tests and experiments and iterations between designers and different types of users.

Only in this way, it is possible to avoid the risk of invalidation of the proposed methodology simply because of possible differences between designer setting and user expectations. Also, a more effective use of the offline instrument and a more accurate validation and possible adjustment of the instrument will be possible.

These steps of development are being performed in parallel with the development of the GUI and offline tools. In general, the overall methodology, including the online tools described above, will be further developed and proposed as innovative means of improvement for improving robustness and resilience of critical infrastructures.

Acknowledgments The research leading to these results has received funding from the European Commission Seventh Framework

Programme (FP7/2007-2013) under grant agreement no. 225553 Project INSPIRE (INcreasing Security and Protection through Infrastructure REsilience). The authors would like to thank Selex Sistemi Integrati and Thales Communications France S. A. partners of the INSPIRE Consortium which have widely contributed to the achievement of INSPIRE outcomes. The authors would also like to thank other persons involved in the Project, namely Luigi Romano (Consorzio Interuniversitario Nazionale per l'Informatica), Iñaki López (S21sec Information Security Labs), Abdelmajid Khelil and Neeraj Suri (Technische Universität Darmstadt), Walter Schmitz (Centre for European Security Strategies).

References

Bevan N, Macleod D (1994) Usability measurement in context in behaviour & information technology 13(1 and 2):132–145

Cacciabue PC, Martinetto M (2006) A user centred approach for designing driving support systems: the case of collision avoidance. Int J of Cogn Technol Work (CTW) 8(3):201–214

Choras M, Stachowicz A, Kozik R, Flizikowski A, Renk R (2009) Ontology-based approach to SCADA systems vulnerabilities representation for CIP. Electronics 11:35–38

MSHTML references http://www.msdn.microsoft.com/en-us/library/aa741317.aspx

Introduction to ActiveX Controls http://www.msdn.microsoft.com/en-us/library/aa751972%28VS.85%29.aspx

D'Antonio S (2010) "INSPIRE: increasing security and protection through infrastructure resilience". ESCoRTS workshop, Ispra, Italy, April 28

Douceur JR (2002) The sybil attack. In: Proceedings of 1st international workshop on peer-to-peer systems (IPTPS)

Germanus D, Khelil A, Suri N (2010) Increasing the resilience of critical scada systems using peer-to-peer overlays. In: Proceedings of the 1st international symposium on architecting critical systems (ISARCS)

ISO–International Standard Organisation (1993) Guidance on usability. ISO/DIS 9241-11

ISO–International Standard Organisation (1999) Human-centred design processes for interactive systems. ISO/FDIS 13407(E)

Khelil A, Jeckel S, Germanus D, Suri N (2010) Towards benchmarking of p2p technologies from a scada systems protection perspective. In: Proceedings of the 2nd international conference on mobile lightweight wireless systems (MOBILIGHT)

Kozik R, Choras M, Holubowicz W (2010) Fusion of bayesian and ontology approach applied to decision support system for critical infrastructures protection. In: Proceedings of MOBILIGHT (Mobile Lightweight Wireless Systems), Barcelona, May

Maymounkov P, Mazières D (2002) Kademlia: a peer-to-peer information system based on the XOR metric. In: 1st international workshop on peer-to-peer systems (IPTPS'02)

The WebKit Open Source Project http://www.webkit.org/

Geko https://www.developer.mozilla.org/en/Gecko

Opera Presto http://www.dev.opera.com/articles/view/presto-2-1-web-standards-supported-by/

Konqueror http://www.konqueror.org/features/browser.php

Java http://www.java.com

VBScript http://www.msdn.microsoft.com/en-us/library/t0aew7h6%28VS.85%29.aspx