ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Proceedings of the
13th European Conference on Cyber Warfare and Security

The University of Piraeus
Greece
3-4 July 2014

Edited by
Andrew Liaropoulos and George Tsihrintzis

acpi

A conference managed by ACPI, UK

Proceedings of the 13th European Conference on Cyber Warfare and Security ECCWS-2014

The University of Piraeus
Piraeus, Greece
3-4 July 2014

Edited by
Andrew Liaropoulos
and
George Tsihrintzis

Papers have been double-blind peer reviewed before final submission to the conference. Initially, paper abstracts were read and selected by the conference panel for submission as possible papers for the conference.

Many thanks to the reviewers who helped ensure the quality of the full papers.

These Conference Proceedings have been submitted to Thomson ISI for indexing.

Further copies of this book and previous year's proceedings can be purchased from http://academic-bookshop.com

E-Book ISBN: 978-1-910309-25-4
E-Book ISSN: 2048-8610
Book version ISBN: 978-1-910309-24-7
Book Version ISSN: 2048-8602
CD Version ISBN: 978-1-910309-26-1
CD Version ISSN: 2048-8629

Published by Academic Conferences and Publishing International Limited
Reading
UK
44-118-972-4148
www.academic-publishing.org

Improving Cyber-Security Awareness on Industrial Control Systems: The CockpitCI Approach

Tiago Cruz[1], Jorge Proença[1], Paulo Simões[1], Matthieu Aubigny[2], Moussa Ouedraogo[3], Antonio Graziano[4] and Lasith Yasakhetu[5]
[1]University of Coimbra, Portugal
[2]iTrust Consulting, Luxembourg
[3]Centre de Recherche Publique Henry Tudor, Luxembourg
[4]Selex ES, Italy
[5]University of Surrey, UK
tjcruz@dei.uc.pt
jdgomes@dei.uc.pt, aubigny@itrust.lu
psimoes@dei.uc.pt, aubigny@itrust.lu
aubigny@itrust.lu
moussa.ouedraogo@tudor.lu
antonio.graziano@selex-es.com
s.l.yasakethu@surrey.ac.uk

Abstract: Originally isolated by design, Critical Infrastructures (CI) based on Industrial Control Systems (ICS) – such as SCADA (Supervisory Control and Data Acquisition) systems - were born within the scope of industrial process control technologies. Having evolved from proprietary systems, ICS eventually started adopting open architectures and standards, becoming increasingly interconnected with existing corporate networking infrastructures and even the Internet. However, as these systems overcame their isolation and moved towards interconnected topologies, they also became more exposed to threats that weren't even remotely conceivable when they were first designed. Particularly, cyber-threats are one of the most significant problems that modern ICS face, as the shortcomings and vulnerabilities of the decade-old ICS technology – some of them known for a long time, but mostly downplayed due to the isolation of such systems – become serious threats that can ultimately compromise human lives. As the security needs ICT and ICS domains cannot be addressed in the same way, this calls for a domain-specific approach to cyber threat detection, designed from scratch to address its particular needs. It must consider the particular characteristics of each networking context, be it ICS or ICT, in order to provide real-time cyber-security awareness for the security teams operating in the control room – this is one of the most important contributions of the CockpitCI FP7 project (http://CockpitCI.eu), which aims at improving the resilience and dependability of CIs. In this paper we present the CockpitCI cyber-detection and analysis layer, also including a detailed description of its most relevant components in terms of role, integration and remote management. This paper will also show how the proposed solution might be effective in dealing with such cyber-threats, by presenting relevant examples.

Keywords: critical infrastructure protection, industrial control systems, SCADA systems, IDS

1. Introduction

SCADA (Supervisory Control and Data Acquisition) is the commonly designation which is used to refer a set of technologies, protocols and platforms used in Industrial Control Systems (ICS). Such systems are used in several scenarios, such as production lines automation, for controlling nuclear or thermoelectric plants, for distribution grids and many other applications.

As their scope was originally restricted to isolated environments, SCADA systems were relatively safe from external intrusion. However, as architectures evolved SCADA systems started to assimilate technologies from the Information and Communication Technologies (ICT) world, such as TCP/IP and Ethernet networking. This trend, together with the increasing adoption of open, documented protocols, exposed serious weaknesses in SCADA architectures. Moreover, the interconnection of the ICS network with organizational ICT network infrastructures, and even with the exterior (for instance, for connection with internal company systems or for remote management) brought a new wave of security problems and attacks to such an extent that the number of externally initiated attacks on ICS systems has increased significantly, especially when compared with internal attacks (Kang 2011).

This situation, together with inadequate systems lifecycle management procedures, disregarding regular updates or patching (Krutz 2006), increases the probability of a successful attack. While such procedures are

trivial matters which are part of the regular maintenance routine in the ICT world, they must be dealt in an entirely different way when it comes to ICS, mainly for two reasons: the fact that some components have to work on a continuous basis without interruptions, up to the point of working years without being reinitialized (ESCoRTS 2010) (Zhu 2011); due to the fact that any software release must be carefully tested by equipment manufacturers before being released, or even due to end-of-life support for specific devices or software frameworks.

Therefore, ICS constitute a critical and strategic asset that is being increasingly targeted by malicious attacks, with potentially catastrophic consequences. In this context, CockpitCI (CockpitCI) is focused on improving the resilience and dependability of CIs.

The idea behind the CockpitCI project is to allow the community of CI (Critical Infrastructure) owners to exchange real-time information about attacks (and tentative attacks). In this scope, each CI incorporates its own real-time Distributed Monitoring System and Perimeter Intrusion Detection System (PIDS). These systems are able to aggregate the filtered and analyzed information of potential cyber-attacks against systems used to support the operation of CIs and identify the potential unsecured area of the CIs.

This PIDS, which is the focus of this paper, performs many of the tasks traditionally associated with a Distributed Intrusion Detection System, with support for diversified and closely integrated detection and analysis techniques and tools. Each PIDS is to be deployed in the targeted area of a CI, in order to detect coordinated cyber-attacks and in order to deploy prevention strategies of isolation.

Through coordinated PIDS operations, it is possible to put in place a specific perimeter to detect potential coordinated cyber attacks on CIs for each type of detected attacks or for mixed cyber attacks. The CockpitCI PIDS is based on a state-of-the-art distributed intrusion detection architecture encompassing a set of detection agents that feed real-time and soft real-time automated correlation and anomaly detection mechanisms, orchestrated together by a management platform.

In this paper the main aspects of the proposed PIDS architecture are presented. Section 2 discusses the problem of security in ICS/SCADA. Section 3 introduces the CockpitCI architecture. Section 4 presents the proposed PIDS, with sections 5 and 6 discussing analysis components and detection capabilities, respectively. Section 7 details integration of eventing and management interfaces. Finally, section 8 presents conclusions and gives some insights into what future directions the project might take.

2. A brief overview of ICS/SCADA security issues

The development of the CockpitCI PIDS architecture was preceded by a requirements analysis phase, with the purpose of understanding the specific characteristics and differences between ICS and conventional ICT infrastructures, from a security standpoint. This study revealed several and significant differences between ICT and ICS domains that are deeply rooted in their own particular characteristics, down to the fundamental priorities that define which are the most important operational and functional properties of the system.

When it comes to their fundamental governing principles, ICS and ICT infrastructures have an inverted set of priorities, a situation that is one of the main causes of SCADA infrastructure security problems. This is partly due to the fact that, as the ICS paradigm evolved, it was not accompanied by an equal progression in terms of an industry mindset that remained unchanged almost since the inception of such technologies. As a result, and due to its critical nature, ICS operation and design practices frequently privilege availability and reliability over confidentiality and data integrity – a perspective that is quite the opposite from the ICT philosophy, which privileges confidentiality and security, followed by communications integrity and, finally, by availability (ISA-99.00.01). This contrast explains why it is frequent to find a lack of mutual understanding between the control systems teams and the security IT staff, within the same organization.

The differences between the ICT and ICS contexts also mean that there is no “one size fits all” solution when it comes to chose and implement security mechanisms. Despite this, importing solutions from the ICT world is often a necessity, which might lead to undesirable side-effects. The fundamental premises for ICT security tools and commonplace lifecycle management procedures, such as patching and updating a system, can become troublesome in an ICS, when faced with situations such as the impediment / high cost of stopping

production or even the explicit prohibition by the system's manufacturer. As an example, a SCADA system customer may be unable to install an update on an operating system unless the manufacturer certifies their software for the update.

Product lifecycle is another matter that separates ICT from ICS systems, with the former having substantially shorter lifecycles, when compared with the latter. In ICS it is frequent for mature systems to be kept in operation, sometimes far beyond their projected lifetime – the "if ain't broke, don't fix it" philosophy. This limits the possibility of implementing some security mechanisms due to the limited capabilities of existing equipment (Igure 2006).

Moreover, SCADA communication protocols, which are responsible for the interaction between field devices, such as PLC (Programmable Logic Controllers) or RTU (Remote Terminal Units) components and the stations that control and monitor them, pose security concerns. One of such examples is the Modbus protocol (Modbus 2006), originally developed by Modicon (currently part of the Schneider Electric Group) in 1979. Modbus is one of the most popular protocols for SCADA applications, thanks to its simplicity and ease of use. However, Modbus suffers from security problems: the lack of encryption or any other protection measures exposes it to different vulnerabilities (Triangle 2002) – if we take into consideration that it is not uncommon to find situations where the ICT and ICS networking contexts are blended within the corporate network, it becomes clear to which point some ICS are vulnerable. Despite this, protocols such as Modbus have a large lifespan and are still being massively deployed and used.

Simply put, when it comes to ICS, technology and platform maturity are valued as an implicit recognition of value and reliability, and even the disclosure of security issues related to them seems to have no effect in discouraging their usage or prompting the adoption of countermeasures to protect them. This has become the root cause of many ICS security issues that have been exploited with a variable degree of success, in recent times, such as the Stuxnet Trojan (O'Murchu 2011).

3. The CockpitCI project

To improve the resilience and dependability of Critical Infrastructures (CIs), the CockpitCI project proposes an architecture that is divided into several modules/components, whose interaction is illustrated in Figure 1 (CockpitCI 2013). Not only does this architecture aim to detect cyber threats using novel strategies for intrusion detection, along with devices specially conceived to monitor CI ICS/SCADA systems, but it also accounts for communication between multiple interdependent CIs, using a Secure Mediation Network (SMN) to share operational and security information.

Among the multiple components that make up the CockpitCI architecture, the Dynamic PIDS provides the core cyber-analysis and detection capabilities, being responsible for continuously assess and protect the electronic security perimeter of each CI. The automatic analysis and detection mechanisms for each PIDS are fed by several field adaptors and detection agents deployed within each CI, which constitute its "eyes", providing the basic information from which the ongoing security status of the CI is inferred. Also, the PIDS encompasses semi-autonomous reaction capabilities, being able to deploy and activate countermeasures, in line with predefined security reaction policies.

For each CI, an Online Integrated Risk Predictor (IRP) module works as a decision support system for management teams, feeding operational indicators (such as process-level data, gathered using the SCADA TLC and ELE adaptors shown on Figure 1) and cyber-security information (generated by the PIDS) into a set of modelling tools, to assess and predict propagation and threat levels for potential cyber attacks on the CI. In this scope, SCADA adaptors translate SCADA data from various components into a common data format, enabling the use of devices from different vendors and legacy SCADA HW/SW (Hardware/Software) while sharing data with the detection layer and the IRP.

The Secure Mediation Network (SMN) provides the means for exchanging security information between CIs, also enabling the use of risk prediction and other analysis mechanisms to assess threats in a global scale, accounting for CI interdependencies (as exemplified by Figure 1, which represents 2 kinds of CIs that are frequently dependent on each other: Telecommunications and Electric Power Distribution).

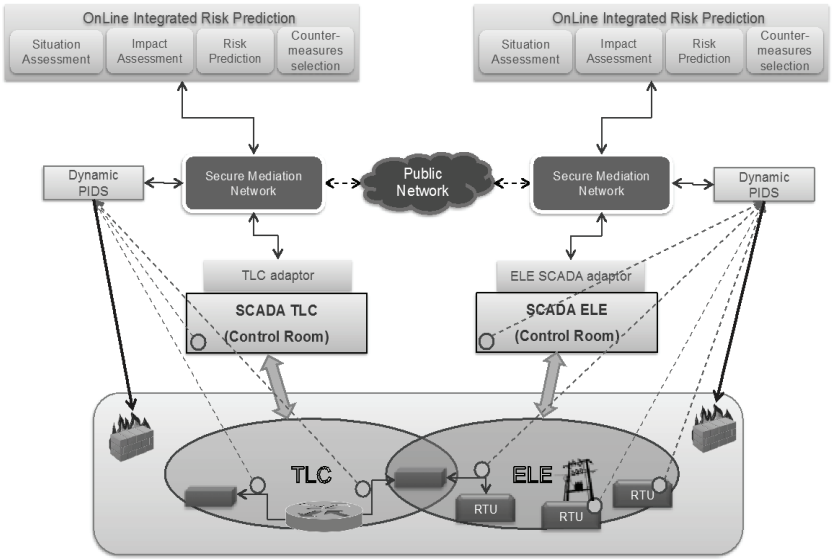

Figure 1: The CockpitCI general architecture (CockpitCI 2013)

Among these components, the CockpitCI PIDS cyber-analysis and detection architecture constitutes the main subject of this paper. The next sections will focus on the presentation and analysis of the basic components of the PIDS, explaining how they work together to continuously monitor and analyze the ICS/SCADA system components of a given ICS in order to perform threat detection. Also, the event correlation and anomaly detection mechanisms that constitute the cyber-security analysis capabilities of the PIDS will be addressed, together with the description of the agents and probes that feed such them with information about ICS/SCADA system's state and its data flows.

4. The CockpitCI cyber-detection and analysis layer within the PIDS

The CockpitCI PIDS incorporates several advanced real-time detection and analysis mechanisms, integrated to constitute a cyber analysis and detection layer for the CI, as shown on Figure 2. It is structured along the three different zones of the CI, each one with its own internal security perimeter: the Field Network, SCADA Process/Operations Network and the IT (Information Technology) Network. This distinction confers the PIDS the ability to deploy agents and security policies customized to the specific needs and characteristics of each network scope.

This architecture was designed to deal with several attack scenarios, from known threats to rogue events, such as: man-in-the-middle attacks, device impersonation, non-authorized tampering, worms, trojans, denial-of-service attacks or flooding, among others. For this purpose, the PIDS is designed in such a way that it integrates different detection strategies, distributed along different levels, namely:

- **Detection agents and field adaptors**, including agents, adaptors and extensions for existing system components, as well as specialized network probes and honeypots (Spitzner 2002) to be added to the network which are able to capture behaviour or traffic patterns (as performed by NIDS – Network IDS components) as well as host (using tools such as HIDS/Host IDS, or antivirus software) and field device monitoring.
- **A distributed multi-zone, multi-level correlation** structure that processes the information provided by the security sensors, **complemented by machine-learning capabilities**, in the form of One-Class Support Vector Machine (OCSVM) (Ma 2003) anomaly detection module, based on adaptive machine learning.
- **Aggressive usage of topology and system-specific detection mechanisms**, based on the fact that the role and behaviour of each system component in an ICS are expected to be more consistent over time than on other types of networks, analysis components are fed with knowledge provided by a number of system specific sources, such as topology databases, policy databases, and trust-based mechanisms, as well as strategically placed honeypots.

The operation of the PIDS components is orchestrated through a Security Management Platform (SMP), which is responsible for managing all the involved components of the solution (see Figure 2). It includes the

mechanisms for managing the security and components of the infrastructure. The SMP is responsible for the maintenance and management of monitoring probes such as IDS and the analysis components, also including monitoring of in-place security and vulnerabilities within the network as well as the maintenance of the latter. Therefore, the SMP has a dual role, dealing with both security audit and maintenance mechanisms.

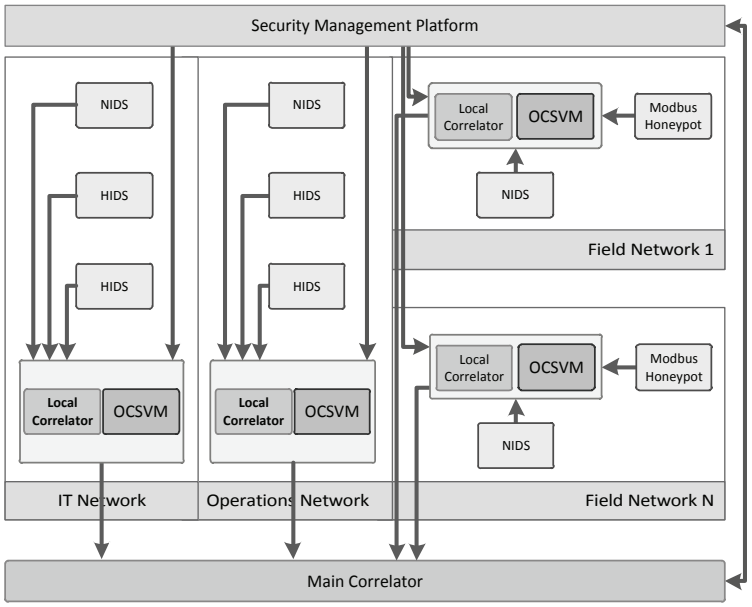

Figure 2: The CockpitCI cyber-detection and analysis layer (red flows=management, green=eventing)

The SMP performs the configuration of detection agents on the field, allowing to set-up their detection thresholds and other relevant parameters. This detection threshold depends on both the risk level of the overall infrastructure (the level of detection shall be higher if the probability of an attacks is higher) and on the specific detection needs (e.g. in case of abnormal event detection on specific system, the SMP shall be able to verify all similar components in the CIs to check their security level).

Due to the demanding availability requisites and little tolerance to delays, the detection architecture is to be implemented using a network that is separate from the SCADA system network (eventually it can use the same physical network, using VLAN (Virtual Local Area Network) or other types of overlay techniques for traffic separation), in order to guarantee that it does not interfere with the normal operation of the control network.

5. Analysis layer

The analysis components of the PIDS provide a way to extract information from the data collected by the agent layer or directly from network traces. These components are arranged in a two-level architecture with local instances fine-tuned for each network scope.

5.1 Local and global correlators

Local correlators perform the first step of correlation, filtering and reducing the number and noise of the alarms generated by the detection layer, while providing a mechanism for security event generation that is able to filter, process and relate events within a network segment (e.g. alarms generated by two or more detection agents, multiple events from the same source).

Local correlators receive the events from local detection agents (e.g. HIDS) on their network scope and process them accordingly with a set of rules, forwarding significant results to a global correlation engine. This approach provides context separation, at the same time allowing for better efficiency and scalability for real time event processing. After local correlation, events are sent to the global correlators and from the latter to the SMN, using the Intrusion Detection Message Exchange format (Debar 2007). IDMEF defines an experimental standard for exchanging intrusion detection related events. As a standard, it can be used as a vendor or product independent enabling intercommunication between different agents such as NIDS or Honeypots.

As illustrated by Figure 2, local correlators receive events from the different agents such as NIDS, HIDS, Honeypots, among others. These agents are distinct according to network zone in which the local correlator is located. Despite of the range of different agents, the local correlator should use the same interface for all of them, as messages are received through an Event Bus (discussed on Section 7). This interface will allow subscribing to the events published by the agents. Local correlators also have an agent adaptor interface that allows for management, via the SMP.

Regarding the event interfaces for the main correlator we have different types: one to receive events from the local correlators and another one to send events to the SMP, both using an Event Bus. As local correlators have already previously processed received events, the main correlator can focus in Multi-Step, Attack Focus Recognition correlation, as well as Alert Prioritization. A management adaptor provides the interface for the SMP to configure the correlator (see Figure 3).

The correlators are implemented using the Esper (Esper) Complex Event Processing (CEP) tool. This was due to the fact that Esper is a multi-platform, flexible and mature tool, in development since 2006. Also, performance tests have shown Esper to exhibit a good balance between memory usage, CPU usage and execution time, when processing hundreds of thousands of events.

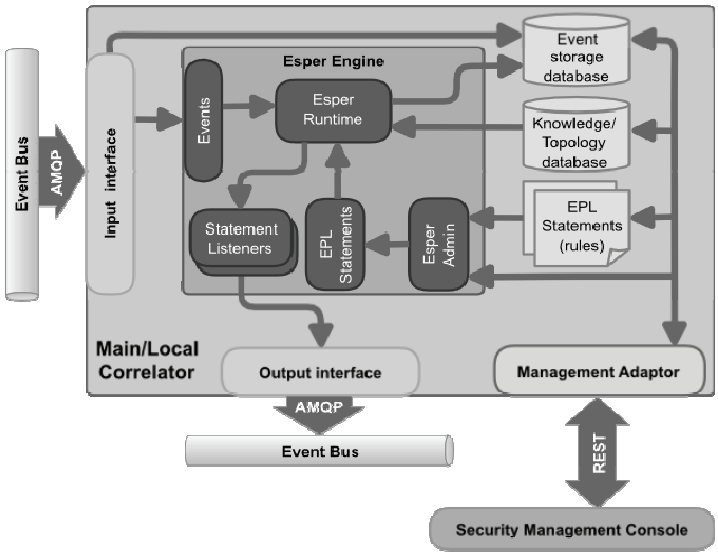

Figure 3: verview of the correlator architecture

Esper can natively accept events represented in XML, among others, which is useful as IDMEF, used by the PIDS, is an XML-based format. If a XML schema document (XSD file) is provided Esper can read the schema and properly present event type metadata and validate statements that use the event type and its properties. To access the elements of the event the correlator uses XPath expressions. If a schema for the XML is provided, the XPath expression needed to reference the attribute can be inferred automatically. Otherwise, expressions can be manually configured.

An overview of the architecture of a PIDS correlator, based on Esper, is pictured in Figure 3. The events received from the input adaptor are sent to the Esper Runtime (EPRuntime); this provides the interface to the event stream processing runtime services. The statements are registered in the EPRuntime and represent the event stream queries and/or event pattern. Each statement can have one or more Statement listeners bound to them. When the condition of a query is verified Esper can trigger the listener(s) bound to the rule, insert the result of the statement into another stream (that already exists or is created at that time) or do both options. If a rule generates a new event, that need to be sent by the correlator to the Event Bus, the listener will interface with the output adaptor to send it. Output events are generated by rules making use of input events, cached events, the internal state and information from external sources.

Esper statements are added to the EPRuntime through the Esper Administrator (EPAdministrator) module. This is an administrative interface to the event stream-processing engine. For security auditing purposes the correlator will log all events and traces of the actions performed to persistent storage. The events will be logged as they are received in the correlator and the EPRuntime shall also log the actions executed by the

correlator. Correlation can make use of information taken from external sources. These sources can provide additional information related, among others, to the definition of the network topology and other detailed system information. These external sources (knowledge/topology databases) can be queried directly from an EPL statement. New rules can be added to the correlation engine dynamically, without restarting the engine.
Using the same correlator tool for the two levels of correlation provides uniformity, since the same language is used to express the correlation operations, and allows easier integration with the Event Bus, as the same interfaces can be used for the two levels. Using the same rule description language for both correlators simplifies the task of rule management by operators and security experts. Additionally, some correlation rules can be used in both correlators without the need to be converted.

5.2 One-Class support vector machines (OCSVM)

OCSVM (One-Class Support Vector Machine) are a natural extension of the support vector algorithm to the case of unlabelled data, especially for detection of outliers. However, unlike SVM or any another classification algorithm, OCSVM does not need any labelled data for training or any information about the kind of anomaly is expecting for the detection process. OCSVM principles have shown great potential in the area of anomaly detection (Ma 2003, Li 2003, Schölkopf 2001). Moreover, OCSVM is capable of handling multiple attributed data (Hsu 2003, Wang 2004), which is well suited for SCADA systems.

The advantages of the OCVSM component are manifold: since OCSVM does not require any signatures of data to build the detection model it is well suited for anomaly-based intrusion detection in SCADA environment; since the detection mechanism does not require any prior information of the expected attack types, OCSVM is capable of detection both known and unknown (novel) attacks, besides being robust to noise in training sets. Also, algorithm behaviour can be controlled and fined-tuned by the user to regulate the percentage of anomalies expected (thresholds, as defined via SMP via the OCSVM management adaptor).

OCSVM operation consists of 2-steps, namely: training and testing. During the training stage OCSVM builds a model from training on normal (i.e., obtained from a system operating under normal conditions, without any attack in progress) data and then classifies the new data as either normal or attack based on its geometrical deviation from the training data in the testing stage. Since the OCSVM detection approach is robust to noise samples, the training data set can include some noise samples (i.e. data which does not correspond to the normal behaviour). An OCSVM component is deployed in IT, Operation and Field network zone(s), therefore requiring different training sets.

Once the training phase is complete, the OCSVM module is capable of detecting possible intrusions (abnormal behaviour) to the SCADA system, based on real-time capture of network traffic traces. The detection module will classify each event whether it is a normal event or a possible intrusion. This information will then be encoded in an IDMEF message and sent to the main correlator, using an adaptor for the Event Bus, in order to react accordingly to the detected intrusions.

6. Detection agents

The detection agents are the lowest level of the detection layer. Their purpose is to gather information from the system. As the format of information provided depends on the type of detection agents used (type of probe), adaptors allow the acquisition of data from the system in a recognised format. Detection agents and adaptors are essential to feed the local correlators of the detection layer with input data regarding suspicious activity. The PIDS encompasses several kinds of probes and detection agents, among which the most relevant are next described.

6.1 Threat detection agents

Network IDS: the perimeter for each network scope is monitored using NIDS components for each one: IT Network NIDS, Operations Network NIDS, and Field Network NIDS. These have interfaces to report the security events to the zone correlator within their network scope. In the PIDS, Snort (Snort) is used for this purpose, albeit other NIDS could be used.

Host IDS: the Host IDS is deployed in the hosts/servers of the system. It is capable of reporting anomalous behaviour in the machine where it is deployed. In the CockpitCI PIDS, OSSEC (OSSEC) is used for this purpose, but other HIDS could be used.

Honeypots: acting as decoys and being capable of detecting attackers probing the network, honeypots provide another source of data for correlation. There are three types of honeypots in the detection layer: IT Network, Operations Network and Field Network honeypots (Simoes 2013).

Exec Checker (linux hosts): capable of detecting malicious network frames by sniffing the traffic, the Exec Checker (in active or passive mode) captures the different parts of an executable in the network traffic to recreate the file and send it to an analysis tool.

6.2 Vulnerability detection agents

Output Traffic Controls (linux hosts): capable of detecting Remote Access Trojans, this specific tool regularly scans system components to check if a remote access toolbox has been installed on components to facilitate external attacks.

Vulnerability Checker (windows hosts): this tool provides a regular control of system vulnerability to check if the monitored systems are vulnerable or not according to an updated database. This tool can be customized for IT or SCADA host profiles.

Configuration Checker (linux/windows hosts): this tool provides a regular control of system configurations to check for unauthorized modification.

6.3 Security event detection agents

Behaviour checker (linux/windows hosts): capable of detecting attacks/threats by analyzing low-level hardware/software behaviour, this specific family of detection agents retrieves hardware/software information such as temperature and CPU (Central Processing Unit) activity in order to avoid accidental or malicious outage.

Security events generated by detection agents are encoded using the IDMEF format. All detection agents have a separate channel (another interface or secure channel) for management purposes, enabling the security staff to adjust the configurations with the scenario requirements, via the SMP. The detection agents send their messages by means of an Event Bus described in Section 7, which also details the management interfaces for the agent adaptors. These interfaces (eventing and management) were designed to ease integration of several types of detection capabilities (such as antivirus, for instance) providing wrapper components for event generation and the management API.

7. Interfaces and integration

This chapter describes the transport mechanisms and interfaces for event data flowing between the several existing components of the PIDS, also addressing their management interfaces.

7.1 The event bus

The Event Bus is the component responsible to manage the communication of the events between the different elements of the PIDS, whose architecture is detailed in Figure 4. Events generated by the different agents within each zone are sent to an Event Bus broker. The broker is then responsible to route this events to a queue from which the local correlator can consume them. After processing and correlating, the events each local correlator sends the events to another broker that feeds the main correlator. The events produced by the main correlator are sent to the main broker that routes them to a queue where they can be sent to the SMP.

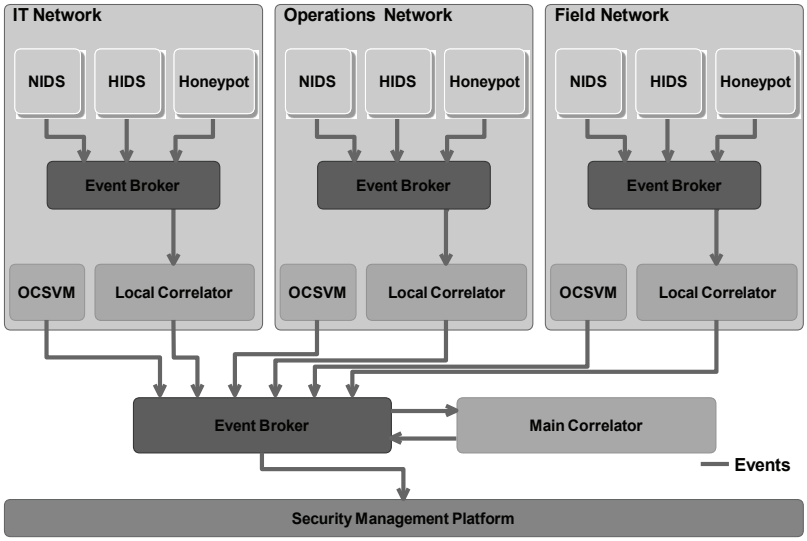

Figure 4: Event bus architecture

The Event Bus uses a Message Oriented Middleware (MOM) (Banavar 1999) to provide efficient event communication among the (sometimes, heterogeneous) components that comprise the PIDS. Several MOM implementations depend on a Message Queue (MQ) system to allow asynchronous message delivery, by providing a temporary storage, on memory or disk, for the messages. Messaging applications communicate with each other through a messaging system, acting either as a message producers (senders) or consumers (receivers). Producers and consumers are loosely coupled, being connected through virtual channels called publish-and-subscribe (one-to-many) channels or point-to-point (one-to-one) channels (Chappell 2004).

For the integration of eventing interfaces, the CockpitCI PIDS adopted an Event Bus based on the Advanced Message Queuing Protocol (AMQP) (OASIS 2011), a wire-level, open standard application layer protocol for MOM that defines a neutral (IDMEF-compatible) encoding scheme of byte sequences to pass over the network. An AMQP messaging system comprises three main components: publisher(s) (which assemble messages and send them to a message queue) consumer(s) (which receive messages from a message queue) and broker(s)/server(s) (responsible for receiving messages from publishers and route them to the right consumers).

The AMQP-based MOM brings a set of important features to the PIDS architecture, namely:

- **Security:** it supports authenticated and/or encrypted transport, using Transport Layer Security (TLS) or Simple Authentication and Security Layer (SASL), to protect events from tampering and/or eavesdropping.
- **Message reliability:** it can guarantee message ordering using a queuing broker, ensuring that messages are delivered to the receiver in the same order in which the sender sent them, with support for disconnection (messages may be held in a queue for deferred delivery).
- **Resiliency:** message delivery semantics provide a range of delivery options, with special emphasis to the *exactly-once* and *at-least-once* modes. These delivery modes, guarantee the message to arrive to the intended destination no matter what. The messaging provider will retry the delivery of a message upon a delivery failure.
- **Scalability and High Availability:** it provides scalability for the communication system thanks to the publisher-subscriber model. The agents can send events, publishing them to a queue/exchange in the broker, which is subscribed by a correlator to receive the messages. This allows adding additional consumers with ease, for failover or to distribute the correlation load across more than one instance. Also, a group of brokers can be clustered together for high availability and/or scalability/load-balancing.

Moreover, the protocol is vendor-neutral and platform-agnostic. There are several open source implementations for many different programming languages.

7.2 Management interfaces

For each managed entity that does not provide a suitable management interface, a component management adaptor/coupling architecture provides an uniform API and Data Model for each component that does not expose its own native management interface.

The Management Adaptor also embeds an API/Data Model module that is responsible for maintaining its data model (state and semantics) properties and also to provide the web service API interface to manipulate them. Accordingly with the mapping rules from the Abstraction Class, attributes exposed by the API layer might have several properties, defining and describing their access mode (read, write), or data types. The API makes use of REST (REpresentative State Transfer) (Fielding 2000) web services, with security being provided with the help of HTTPS with other authentication mechanisms such as client certificates or signed requests.

The data model structure for management adaptors is standardized, being inspired on hierarchical models usually found on management protocols such as SNMP (Case 2002), being arranged as a tree. Asynchronous events are also supported though inclusion of eventing properties, enabling a specific attribute to generate notifications when its state changes.

8. Conclusion

This paper presents the architecture of the PIDS within the CockpitCI architecture. This architecture was designed to address the special cyber-security needs of CIs, such as ICS/SCADA systems, being based on a distributed approach that attempts to bring the most effective detection mechanisms and tools together with correlation and anomaly detection analysis techniques, in order to create a solution that starts with the state-of-the-art in CI security as its baseline.

A strong point of this architecture lies in its capability for assimilation of a diverse range of detection tools in a coherent framework with homogeneous coordination and orchestration. Using distributed two-level correlation capabilities the PIDS is able to get a micro and macro-perspective on the ongoing status of the monitored CI, while being capable of dealing with unknown threats, thanks to the incorporation of machine-learning anomaly detection features. Future work will address improved integration with the SMN, while expanding on functionality and diversity of detection components.

Acknowledgements

The authors would like to thank the support of the CockpitCI (FP7-SEC-2011-1 Project 285647) and iCIS (Intelligent Computing in the Internet of Services – CENTRO-07-0224-FEDER-002003) projects.

References

OASIS, Advanced Message Queuing Protocol (AMQP), version 1.0, available at: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=amqp, July 2011.

Banavar, G., Chandra, T., Strom, R. and Sturma, D. (1999) A Case for Message Oriented Middleware, IBM T. J. Watson Research Center, Hawthorne, New York.

J. Case et al. (2002) Introduction and Applicability Statements for Internet Standard Management Framework, IETF RFC 3410, December 2002.

D. Chappell (2004) Enterprise Service Bus, O'Reilly Media, 2004

CockpitCI, CockpitCI FP7-SEC-2011-1 Project 285647, available at: http://CockpitCI.eu.

CockpitCI (2013) CockpitCI FP7 Deliverable D3.1, Requir. and Reference Arch. of the Detection Layer.

H. Debar, D. Curry, B. Feinstein (2007) Rfc 4765: The intrusion detection message exchange format (IDMEF), March 2007, http://www.ietf.org/rfc/rfc4765.txt.

ESCoRTS (2010), TAXONOMY of SECURITY SOLUTIONS for the SCADA Sector, Deliverable 2.2,

Esper Complex Event Processing, EsperTech, available at: http://www.espertech.com/products/esper.php.

Fielding, R.T. (2000) Architectural Styles and the Design of Network-Based Software Architectures, Ph.D. Dissertation, University of California, Irvine.

Hsu, C., Chang, C. and Lin, C. (2003) A practical guide to support vector classification, Technical report, Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei.

Igure, V.M.; Laughter, S.A. and Williams R.D. (2006) Security issues in SCADA networks, Computers; Security, Volume 25, Issue 7, Pages 498-506, 2006.

ISA-99.00.01 (2007) Security for Industrial Automation and Control Systems - Part 1: Terminology, Concepts, and Models, American National Standard.

Kang, D. et al., (2011) Proposal strategies of key management for data encryption in SCADA network of electric power systems, Int. Journal of Electrical Power & Energy Sys., Vol. 33, Iss. 9, Nov. 2011.

Krutz, R. L. (2006) Securing Scada Systems, USA: Wiley Publishing, Inc., 2006.

K. Li, H. Huang, S. Tian and W. Xu (2003) Improving one-class SVM for anomaly detection, Proceedings of the Second Int. Conference on Machine Learning and Cybernetics, Xi'an, 2003.

J. Ma and S. Perkins (2003) Time-series novelty detection using one-class support vector machines, Proceedings of the International Joint Conference on Neural Networks, July, 2003, pp. 1741-1745.

Modbus-IDA (2006) Modbus Application Protocol Specification V1.1b.

L. O'Murchu, N. Falliere (2011) W32.Stuxnet dossier, Symantec White Paper, February 2011.

OSSEC, Open Source SECurity, Trend Micro, available at: http://www.ossec.net.

B. Schölkopf, J. Platt, J. Shawe-Taylor, A.J. Smola, and R. Williamson (2001) Estimating the support of a high-dimensional distribution, Neural computation, Vol. 13, No. 7, pp. 1443-1472, 2001.

P. Simões, T. Cruz et al. (2013) On the use of Honeypots for Detecting Cyber Attacks on Industrial Control Networks, In proc of 12th European Conf. on Information Warfare and Security (ECIW 2013).

Snort IDS, Sourcefire, available at: http://www.snort.org.

Spitzner, L. (2002) Honeypots: Tracking Hackers, Addison-Wesley Professional.

Triangle MicroWorks, Inc (2002) DNP3 Overview, Raleigh, North Carolina, http://www.trianglemicroworks.com/documents/DNP3_Overview.pdf.

Y. Wang, J. Wong, and A. Miner (2004) Anomaly intrusion detection using one class SVM, presented at 5th Annual IEEE Information Assurance Workshop, West Point, New York, 2004.

Zhu, B et al. (2011) A taxonomy of Cyber Attacks on SCADA Systems, Proc. of the 2011 Int. Conf. on Internet of Things and 4th Int. Conf. on Cyber, Physical and Social Computing (ITHINGSCPSCOM'11).