Positional: A Dual-layer Collaborative Host-based Architecture for Securing Industrial Networks

Peter M. D. Scully
Department of Computer Science,
Aberystwyth University, Wales, United Kingdom, SY23 3DB

This paper presents a security information distribution architecture that allows decentralised adaptation and response to security threats. The inspiration for this work and its underlying principles lie in the human immune system and its mechanisms for adapting to and sharing information about threats. The architecture assumes that the network is heterogeneous and contains low-level controllers (such as Siemens programmable logic controllers) as well as computers running conventional operating systems. It attempts to reduce computational overhead on components that are limited in resources and to exploit computational capacity where there is excess. In part this is achieved by using a set of metrics to measure damage that are analysed locally and distributed. Collated input data is forwarded by less capable devices and processed by higher performance (trusted) components to produce analytical models and recovery models.

Artificial Immune System; Industrial Control Systems; Host-based; Collaborative; Security Information Architecture

1. INTRODUCTION

In this paper we introduce an architectural solution to address impracticalities in patching remote or poorly connected industrial networks and the problem of the dynamic ecosystem of threat and vulnerability types, especially those giving opportunity to advanced persistent threats (APTs) that can affect the operation of industrial control systems (ICS).

The Open Source Vulnerability Database (OSVDB) reported a rising trend in SCADA specific vulnerabilities since 2007 with Siemens, the programmable logic controller (PLC) manufacturer with the largest market share, hosting the majority of vulnerabilities in the category, Kouns and Martin (2015). One vulnerability example is the ICS-ALERT-11-186-01 security bulletin issued in 2011. It described how password protection could be bypassed or disabled on Siemens SIMATIC S7 200, 300, 400 and 1200 series PLCs ICS-CERT (2011). To this day, only 1200 series PLCs have been patched. Until as late as May 2007 firmware upgrades ($\leq$v2.6.0) in the Stuxnet-targeted S7-315 series PLCs required a physical visit to each device to complete the operation. This method can be infeasible for PLCs operating independently in remote locations.

Simultaneously we find complex hiding strategies in malware and APTs, such as packing, fuzzing and obfuscation. Various open source libraries and search tools have been released to aid security researchers with fingerprinting, exploit identification and interoperability between bespoke software and the PLCs. Combined, this makes creating new APTs and writing scripts to control PLCs even easier while making pre-emptive detectors more specialised or harder. ICS cyber security standards such as National Institute of Standards and Technology's (NIST) 800-82 rev 2 Stouffer et al. (2015) describe a best practice framework and guide us toward a barrier-based defence-in-depth strategy. However, against the malware described above and in the cases of Stuxnet in 2009-10 Falliere et al. (2011) and the Idaho National Labs Aurora APT attack types in 2007 (Knapp 2011, p37), a network border barrier-based defence is ineffectual.

The inspiration for our host-based solution and architecture's design is drawn from the innate and adaptive human immune system Murphy et al. (2012). Our holistic view of the cells and interactions of this complex system give us infrastructural principles that we have applied to this design. These include strategic locations for many regional decentralised hubs of adaptive cells, as exhibited in lymph nodes. Local decisions guided by signalling and based on timely localised and contextual information, as well as directed by prioritised local objectives and on system-wide survival objectives, as found in immune responses and in sickness behaviours.

As shown in section 2.1, these, among other immunity-inspired principles and lessons from earlier work Scully (2016) lead to a collaborative, host-based and hybrid-decentralised architecture. Within this paper, we will summarise the design and discuss how to apply this architecture to industrial networks.

2. ARCHITECTURE DESCRIPTION

2.1. Design Principles

The design principles satisfy two categories of system goals, self-healing and self-management.

Self-Healing:

Detection: devices use trained models to recognise worsening metric scores or deviations from normal activity. **Recovery:** devices execute recovery modules to revert to normal or an improved operational state, guided by historical data and metric scores. **Collaborative Model Training:** more powerful devices will build and distribute detection and recovery models using device activity data and metric scores.

Self-Management:

Redundancy: more powerful devices store recovery tools, historical data and offer resources to support other devices. **Social Sensing:** devices inspect each other, via a request to discover the device's current activity. **Self-Organising:** device roles adjust depending on the available resources. **Optimising Footprint:** devices adapt their processing tasks and communications for minimal impact on system resources. **Regional Clustering:** devices recognise bottlenecks associated to network scaling and cluster nodes into regional sub-architectures. **Collaborative Decisions:** devices request confirmation from other devices before taking a repair or block action on a peer device. **Trusted Communications:** devices will communicate using trusted session mechanisms. **Openness:** devices will enrol newly connected devices into the architecture by installing the client. **Avoidance Strategies:** architecture devices will adapt their behavioural profiles to avoid malicious software targeting the architecture.

2.2. Architecture Overview

A software architecture client runs on all network devices and consists of two processes. A lightweight process (LWP) and an intensive role process (IRP). The LWP monitors device inputs, evaluates performance metrics, executes detection and recovery mechanisms, receives updates and transmits performance summary data to other LWPs and transmits input data to IRP(s). The IRP consolidates and analyses received network-wide data to produce detection and recovery modules. The IRP will then send up-to-date modules to LWP devices. The operable state of the industrial network is preserved by manoeuvring the system state to uphold the performance metric scores.

2.3. Communication Mechanisms

The security information distribution mechanisms used by the architecture include unicast input data transmissions from LWP to IRP devices, multicast transmissions of detection and recovery modules from IRP to LWP devices and broadcasts of performance metric score updates between the LWP devices.

The architecture coordination communications require knowing which devices can perform which roles, this results in a dynamic hybridised peer-to-peer topology. As such, LWP devices poll their peers via multicast and IRP devices register their role within a multicast session. Devices may choose their preferred neighbour IRP(s) by entering and transmitting a request within the IRP multicast session.

2.4. Distribution of Workload

Architects of host-based software intended to run on industrial network devices must make *responsible* use of the resources, particularly when trading with operational risk reduction. The impacts of improved decision accuracy and architectural resilience are key factors in this trade-off. LWP and IRP processes monitor system usage and their own resource usage by employing a system information gatherer and reporter (SIGAR). LWP devices can then feasibly operate on devices such as human machine interfaces (HMIs), PLC modules (described below), data acquisition and historian servers, by choosing when tasks and transmissions are actioned.

The IRP has time intensive processing tasks that are executed in coordination, such as unpacking and consolidating received data and analysing the collated data streams to produce detection data models. To support the execution of IRP tasks we introduce an additional piece of hardware into the IRP device pool, a trusted dedicated server and/or trusted computing cluster, see Fig.1. The data stream analysis task is parallelised by separating and distributing subtasks among the IRP devices; a tool such as *Apache Flink* Alexandrov et al. (2014) at 50 MiB in size is feasible. A publish-subscribe mechanism, such as *Apache Kafka* Kreps et al. (2011) at 11 MiB in size is feasible to collect, transmit the input data at LWP devices and collate those data streams at the IRP devices.

To gain reliable operational sensor information we must execute the LWP client on each PLC rack without disrupting the PLC control. The Stuxnet malware falsified reported sensor values by modifying operational blocks (OB) and data blocks (DB) within the memory of the PLC Falliere et al. (2011). We conjecture that driver sensor value data can be delivered to the architecture, via remote procedure calls, by attaching new hardware on to each PLC rack's communication bus. Such a device, we call a *PLC bus module* (PLCBM), would have its own dedicated processor and network interface, see Fig.1.

2.5. What Data to Collect

We will illustrate each of the performance metrics using the example topology in Fig. 1. Our objective of these selected metrics is to measure damage, physical damage and production output. Implementation of these metrics must be use case specific. Below are the six metric categories that we have selected to indicate and evaluate an automation system's state:

Sensor value anomaly indicators (SAI): are values from automation sensors (1-5) collected by the PLCBM and acquisition server(s) via request-response messaging using SCADA protocols.

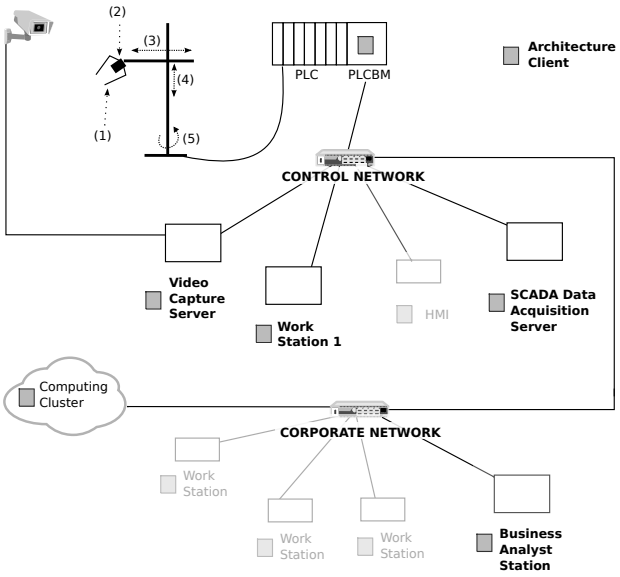

Figure 1: *Toy example of an industrial network topology. Highlighted are added hardware, automation sensors and all devices running the architecture client.*

APT indicators (APTI): monitor activity indicative of connections to/from the network and irregular device activities. These well studied items range from application-layer firewalls, to process call tree analysis, to audit logs and password alerts logs, etc.
PLC state anomaly indicators (PLCI): monitor the PLC state as seen by the PLCBM, acquisition server, and other developer workstations; these indicators include functional block checksums and timestamps.
SCADA network packet indicators (SPI): monitor the communications to the automation system via SCADA network protocol-specific intrusion detection systems and firewalls.
Business success indicators (BSI): measure performance of output quantity, output quality and maintenance costs to indicate undesirable organisational effects caused by a recent system state.
Video capture of device indicators (VCI): monitor the automation system equipment's visual behaviours to indicate anomalies in physical activity.

2.6. Machine Learning

2.6.1. Belief Weights

A rolling mean calculates the belief weighting of each metric as they arrive from a neighbouring device. New metric reporting behaviours are therefore incorporated progressively into the architecture's decision making; thereby managing dynamic shifts in a specific device's behaviour. A binary weight value of 1 is recorded if a neighbour device sends a metric score, otherwise a weight value of 0 is recorded. Fig. 2 illustrates two devices' matrices. In device 1's matrix, device 2 is assigned a weight of 1 for APTI, SPI, BSI and VCI and otherwise 0.

2.6.2. Multi-Layer Classifier

The detection classifier model incorporates new local input data and pre-evaluated network-wide indicator knowledge from other trusted devices.

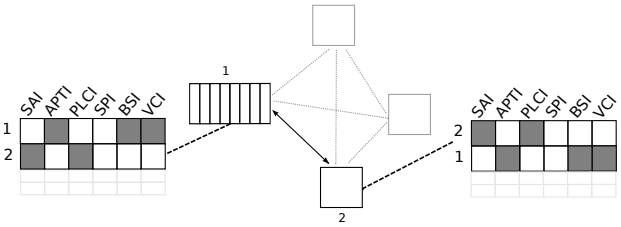

Figure 2: *Example of current metric score matrices per device. White boxes in each matrix indicate available scores. Row order is egocentric, following rows are sorted by age. Devices 1 and 2 highlighted for illustration.*

Each classifier uses up-to-date metric summary scores as inputs. This mechanism requires that all devices broadcast their numerical scores. The local computational effort is low compared to complete source data analysis. Decisions made using this approach do incur the transmission cost of the broadcasts; however, in the event of long periods of disconnection decisions can still be made holding historical logs of metric score matrices. Our approach is faster than request-analyse-response and more reliable under poor connectivity than a request-response pipeline, typical in centralised or hierarchical client-server approaches.

2.6.3. Classifier Fitness Function

Using a data fusion filter, the performance metric scores can be fused to reveal a final evaluation score. The illustrative fitness function in Eq. 1 takes account of initial belief weighting ($w_{i,j}$), a *current trust* variable (t_j), an *egocentric* variable (w_j) and the present score value ($V_{i,j}$) of the metric from the device's matrix of indicator values. m and n are rows and columns of the matrices shown in Fig 2.

$$\sum_{j=1}^{m}\sum_{i=1}^{n} V_{i,j} * w_{i,j} * t_j * w_j \tag{1}$$

The *current trust* variable (t_j) represents intrinsic trust based on behaviour over time. t_j is linearly reduced from 1.0 as a matrix row ages, relative to an error distance from an average delay in receipt per row. The *egocentric* variable (w_j) represents a prioritisation of local over non-local. We arbitrarily limit the egocentric view's effect by reducing linearly the weighting for each matrix row as: $w_j = 1.0, .., 0.75$. $V_{i,j}$ is a positive MIN-to-MAX (from poorest to best) indicator measurement value. $w_{i,j}$ are assigned as described in 2.6.1. Investigation of data fusion techniques, such as Julier and Uhlmann (1997), may be beneficial before implementation.

2.7. Recovery Mechanism Example

The IRP's recovery learning task extracts uploaded binaries and code and data block writes transmitted to the PLCs in an S7 communication protocol packet trace log, with Wiens (2013). This static recovery approach relies on unencrypted automation instruction commands in Siemens S7 transmissions

to many of the S7 PLC models. The extracted and time-series stacked recovery artefacts form the PLC-specific recovery modules with local and system-wide metric scores. Each artefact is ranked by its expected post-recovery evaluation score. IRP will distribute the top ranked parameter change stack and binary checkpoint stack to LWP devices.

An extracted binary, code block or parameter value with data block address is attached to a new S7 transmission with Nardella (2013). Checksum confirmations are then verified. Under a verification failure, an alert is issued and the next artefact is attempted under guidance. Further failures issue a STOP_PLC command and an alert issued.

Recovery operations can be issued immediately, under guidance or autonomously, by storing the most relevant recovery solutions on the PLCBM. In the autonomous case, a recovery artefact can gain an associated probability of success by measuring the similarity of current contextual conditions and performance scores to earlier tracked human-led recovery decisions and their metric score outcomes.

3. DISCUSSION

3.1. Application

The application target of this conceptual architecture is in networked real-time, automated and embedded systems, including automated part assembly lines and life-critical automation processes.

3.2. Challenges

The key challenges of the architecture are in the computational expense, transmission expense, system validation, application-specific tuning and risk reduction in a critical and vulnerable environment. We expect the computational expense has been sufficiently mitigated via load-balancing, i.e. IRP, and in infrastructural design, i.e. added hardware and IRP. The presented architecture employs a redundant peer-to-peer topology operating on real-time distributed data; a transmission heavy approach. While minimising the computational expense on critical devices is essential, this incurs further network transmission cost. The transmission requirements have been reduced by using a minimal representation of state data (pre-evaluated metric scores) for the independent decentralised decision making. Data transmission heuristics specific to decentralised systems Scully (2016) can further reduce this. The feasibility of log data transmissions from LWP to IRP is open. Deriving the semantic time-relevance of values from the real-time VCI and delayed BSI is another challenge to account for before implementing the classifier. A validation methodology of the architecture for application to industrial networks remains open.

3.3. Conclusion & Future Work

This paper has introduced a security information architecture for industrial networks with a design inspired by immunity. The key contributions for security architectures in ICS, SCADA and embedded systems are the multi-layer classifier, evaluation via data fusion and pre-evaluated classifier score transmissions. These simple mechanisms enable decentralised decision making on low-level controllers using network-wide contextual information even under periods of disconnection.

Follow on phases will be a feasibility study of the architecture built for an industrial network testbed focused on the S7 range, using an embedded systems controller as the PLCBM. Then comparative and long term studies on device and network infrastructure impact, in decision accuracy tuning and in recovery evaluations.

REFERENCES

Alexandrov, A., R. Bergmann, S. Ewen, et al. (2014). The Stratosphere Platform for Big Data Analytics. *The VLDB Journal 23*(6), 939–964.

Falliere, N., O. M. Liam, and E. Chien (2011). Symantec Security Response: W32.Stuxnet Dossier. *Retrieved on 2012.07.16*. http://symantec.com/../w32_stuxnet_dossier.pdf.

ICS-CERT (2011, July). ICS-ALERT-11-186-01: Siemens SIMATIC Controllers Password Protection Vulnerability. July 2011. *Retrieved 2015.08.25.*. https://ics-cert.us-cert.gov/alerts/ICS-ALERT-11-186-01.

Julier, S. J. and J. K. Uhlmann (1997). New Extension of the Kalman Filter to Nonlinear Systems. In *AeroSense'97*, pp. 182–193. International Society for Optics and Photonics.

Knapp, E. D. (2011). *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. Syngress.

Kouns, J. and B. Martin (2015). Open Source Vulnerability Database (OSVDB) Search Engine. *Retrieved on 2015.08.25.* http://osvdb.org.

Kreps, J., N. Narkhede, J. Rao, et al. (2011). Kafka: A Distributed Messaging System for Log Processing. In *6th International Workshop on Networking Meets Databases (NetDB)*.

Murphy, K. M., P. Travers, M. Walport, et al. (2012). *Janeway's Immunobiology*, Volume 7. Garland Science New York, NY, USA.

Nardella, D. (2013, September). SNAP7 – Communication suite for natively interfacing with Siemens S7 PLCs. *Retrieved on 2014.11.07.* http://snap7.sourceforge.net/sto.

Scully, P. M. D. (2016). *CARDINAL-Vanilla: Immune System Inspired Prioritisation and Distribution of Security Information for Industrial Networks*. Ph. D. thesis, Aberystwyth University.

Stouffer, K., V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn (2015). Guide to Industrial Control Systems (ICS) Security. NIST 800-82 Rev.2. *NIST Special Publication 1*(800-82), 247.

Wiens, T. (2013). S7comm Wireshark Dissector Plugin. *Retrieved on 2014.11.08.* https://wiki.wireshark.org/S7comm.