Leveraging Virtualization Technologies to Improve SCADA ICS Security

T Cruz, R Queiroz, J Proença, P Simões, E Monteiro

Department of Informatics Engineering
University of Coimbra, Portugal
E-mail: tjcruz@dei.uc.pt; rqueiroz@student.dei.uc.pt; jdgomes@dei.uc.pt*; psimoes@dei.uc.pt; edmundo@dei.uc.pt*

Abstract: *In recent years, Supervisory Control and Data Acquisition (SCADA) Industrial Control Systems (ICS)–systems used for controlling industrial processes, power plants, or assembly lines–have become a serious concern because of security and manageability issues. While the introduction of virtualization technologies has been instrumental in helping ICT infrastructures deal with such problems, their adoption in the ICS domain has been slow, despite recent developments such as the introduction of hypervisors or software-defined networking. This paper provides an overview of the usage of such technologies to improve SCADA ICS security and reliability; it also proposes advanced use cases.*

Keywords: *Virtualization, Critical Infrastructure Protection, Industrial Control Systems*

Introduction

In recent years, SCADA ICS–systems used for controlling power plants, assembly lines, or industrial processes, often part of critical and/or strategic infrastructures–have become a serious concern because of security and manageability issues. After years of air-gaped isolation, the increased coupling of ICS and ICT systems, together with the absence of proper management and security policies (Krutz 2006), disclosed several weaknesses in SCADA ICS, which were left exposed to attacks and potentially catastrophic consequences. These problems hardly constitute any novelty within the ICT domain, which has dealt with them for decades, prompting the development of specific tools and protocols, as well as for the establishment of management frameworks, such as Information Technology Infrastructure Library (ITIL) change management (Galup *et al.* 2009) or security-oriented policies.

However, ICT-specific practices cannot be easily ported to the ICS domain. For ICS operators, equipment manufacturers, and software developers alike, reliability is the top priority. Continuous operation and operational safety targets make it difficult to deploy several ICT-specific strategies and tools because of the potential impact on the ICS. This has pushed the industry, researchers, and standardization organizations to conceive ICS-specific security and management solutions and frameworks, as well as to publish guidelines documenting best practices. New product lines have also been introduced, with added security features and management capabilities.

Still, the ICS paradigm itself remained relatively unchanged, as proposed solutions try to fix what is wrong without attempting to introduce significant change into existing systems. This solution is far from optimal, as typical lifecycle-management operations, such as security patch deployment, are still an issue in modern SCADA ICS, the same being true for change management. In contrast, these issues have been addressed in the ICT domain for years through the continuous development of technologies, tools, and practices designed to address

such needs. Virtualization technologies, which influence ICT computing and communications infrastructures, are among these developments. Developments such as hypervisors, Software-Defined Networking (SDN), or Network Function Virtualization (NFV) are reshaping the ICT ecosystem, providing the means to rationalize the use of computing and communications resources, also being instrumental to optimize and/or to improve aspects such as lifecycle management, energy efficiency, reliability, or security, among others.

From an ICS-security and -reliability perspective, device and infrastructure virtualization may have a similar impact as they had for ICT, as the industry slowly starts to absorb some of the technologies customized and fine-tuned for critical infrastructure environments. However, this process is still in early stages, not only because the specific ICS use cases for several virtualization technologies have yet to be developed, but also because extensive testing is required for its certification in such environments. In this scope, this paper consists of an extended version of an earlier article (Cruz *et al.* 2016)–analysing the application of virtualization technologies for communications and computing resources in ICS contexts, with a focus on recent developments, open challenges, and benefits, from a security and reliability-oriented perspective.

The rest of this paper is structured as follows. The next section discusses the problem of security in ICS/SCADA, also explaining the potential benefits of introducing domain-aware virtualization technologies in such environments. Immediately following is a discussion of the introduction of network virtualization technologies in SCADA ICS and its security benefits. Next, the advantages of introducing partitioning hypervisors in ICS are addressed by describing a virtualized Programmable Logic Controller (PLC-) -use case. Finally, the authors present conclusions and insights about future developments.

Virtualization and SCADA ICS Security

As their scope was originally restricted to isolated environments, SCADA systems were considered relatively safe from external intrusion. However, as architectures evolved, these systems started to assimilate technologies from the 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, a situation that was aggravated by factors such as the use of insecure protocols, including Modbus (Triangle 2002) and inadequate product lifecycle-management procedures (Igure, Laughter & Williams 2006), the latter being responsible for the proliferation of devices and components beyond their end-of-life-support status. Also, the interconnection of the ICS network with organizational ICT network infrastructures, and even with the exterior (for remote management), brought a new wave of security incidents, with externally initiated attacks on ICS systems increasing significantly, especially when compared with internal attacks (Kang *et al.* 2011). Overall, this situation has become the root cause of many well-known ICS security incidents, such as the Stuxnet Trojan (O'Murchu & Falliere 2011).

In fact, ICS security cannot be approached in the same way as its ICT counterpart, as both domains differ significantly in terms of their fundamental design principles. Due to their critical nature, ICS-operation and -design practices frequently privilege availability and reliability over confidentiality and data integrity—a perspective that is quite opposite from the ICT philosophy, which follows an inverse order of priorities (ISA-99.00.01).

The differences between the ICT and ICS domains also mean that there is no 'one-size-fits-all' solution when it comes to choosing and implementing security mechanisms. 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, especially with situations such as the impediment/high cost of stopping production (Zhu *et al.* 2011), or even the explicit prohibition by the system's manufacturer, as any software release has to be certified before being released. Also, several security mechanisms, such as anti-virus software, are frequently ill advised by SCADA software providers, as they might interfere with the response latency of the host. The same rationale applies to anything deployed in the middle of the critical communications path (for example, an inline network Intrusion Detection System), as it may induce latency or some other sort of reliability issue.

Ironically, much of the problems faced by ICS are not entirely new, as they were known well before in the ICT domain, which has undergone several paradigm shifts and undertaken major technological steps to deal with them. More recently, the rise of the virtualization paradigm has become instrumental in changing the ICT computing landscape and providing the means to leverage computing and communications resources through consolidation and efficient management. Technologies such as hypervisors, SDN, or NFV are contributing to rationalizing, streamlining, and reshaping of infrastructures and devices, up to the point of changing the way communications and computing resources are consumed by end-users.

In terms of security and reliability, the impact is manifold. For instance, by creating a virtual machine (VM) snapshot, it is possible to rollback changes in case of failure or corruption caused by a failed OS patch or malicious tampering; VMs can be cloned for sandboxed testing, prior to deployment into production; hypervisors can perform in-place behaviour monitoring of instances for security and safety purposes. Similarly, technologies such as SDN, which constitute a flow-oriented virtualization mechanism for networks, allow for the flexible creation and management of network overlays on top of existing physical infrastructures, while also enabling significant security and reliability benefits (Proença *et al.* 2015). NFV, in its turn, can work together with SDN to virtualize network equipment functionality, spreading it across the communications and computing infrastructure in an efficient and rational way, and also enabling the creation of innovative security solutions designed to better couple with the increasingly distributed nature of modern ICS and associated threats (Cruz *et al.* 2015).

But the introduction of ICT-like virtualization techniques in ICS is not a straightforward process. For operators, equipment manufacturers, and software developers alike, reliability, operational safety, and continuous operation are top priorities, which make it difficult to deploy several IT-specific strategies and tools, because of the potential impact on the ICS. For example, the latency overhead of certain mechanisms may not be compatible with real-time operation requirements. Hypervisors must cope with the (soft) real-time requirements of ICS applications; any attempt to introduce SDN or NFV must account for the potential impact in terms of ICS reliability or latency.

Despite the constraints, the potential efficiency, security, and reliability benefits for ICS are enough to justify the progressive development and introduction of domain-aware virtualization technologies. For instance, real-time hypervisors can provide safe partitioning and isolation, which will enable the creation of managed execution environments for real-time workloads, with continuous assessment of partition behaviour, and also provide rollback capabilities for potentially compromised systems. Use of SDN technologies can provide the ICS operator with the means to monitor the ICS communications infrastructure behaviour, while easing the implementation of countermeasures and deployment of security

mechanisms. As ICS become increasingly distributed, NFV can provide the means to efficiently spread functional security components across the ICS communications and computing infrastructure in order to better couple with the dispersed nature of the protected systems. The next section of this article will discuss how domain-aware virtualization can provide effective security benefits for ICS, with a focus on two major scopes: communications and computing.

Virtualization of SCADA ICS Communications Infrastructures

This section is specifically concerned with the introduction of SDN and NFV technologies within the SCADA ICS scope. For this purpose, the security benefits of the technologies hereby discussed will be analysed from a broad perspective, both in terms of the physical ICS dimension and dispersion of its scope, ranging from plant-level to distributed Industrial Automation and Control Systems (IACS) use cases. All sections will start with a brief introduction of their respective cornerstone concepts, namely SDN and NFV, in order to ease their introduction in the context of SCADA ICS security.

SDN and SCADA ICS

In conceptual terms, network architectures encompass three planes, which represent different areas of operation (Kreutz *et al.* 2014; Ellanti *et al.* 2005), as illustrated in **Figure 1**, below: management, control, and data. In this model (there are other variations), each plane has a specific function in terms of data transmission and network operations.

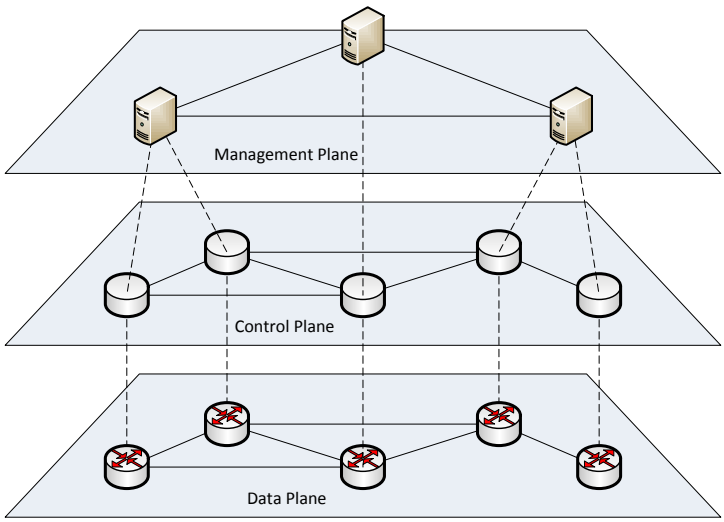

Figure 1. Network planes (adapted from Kreutz *et al.* 2014)

In this model, each plane plays a well-defined role, each one with its own characteristics:

- The **management plane** corresponds to traffic generated by services used for network infrastructure provision, maintenance, and monitoring. Such traffic can be transported through in-band (sharing the same link as user/normal traffic) or out-of-band (OOB) connections (a separate link/connection dedicated for management operations) (Schudel & Smith 2007).
- The purpose of the **control plane** (or signalling plane) is to support the setup of the data plane, including traffic between network elements related with policy or routing information exchanges. This is the case with switches, which may use specific protocols to exchange bridge information among them in order to infer topology information and to avoid loops. Control-plane traffic includes signalling, routing

information, and link-state protocols, among other types of traffic (Schudel & Smith 2007).

- The **data plane** (also referred as the user plane, forwarding plane, carrier plane, or bearer plane) is responsible for carrying user data. Traffic belonging to this plane does not involve source or destination IP addresses belonging to network elements, such as routers or switches,, as it is expected to involve only end devices, such as computers and servers (Schudel & Smith 2007), which use the network for transport purposes.

SDN departs from the vertical integration that is characteristic of the traditional networking model, proposing an architecture that decouples forwarding functions (data plane) and network control (control plane), with the aim of introducing direct programmability into the network, to applications and policy engines alike (Kreutz *et al.* 2014). The control plane is moved outside the forwarding network elements and placed in a logically centralized controller (whose functionality may be spread among several instances, to improve scalability and resilience (Yeganeh, Tootoonchian & Ganjali 2013)), with the data plane remaining in place. The term SDN (for 'Software Defined Networking') was first introduced in an article (Greene 2009) referring to the Openflow project (ONF 2012) at the time being developed at the University of Stanford, which eventually became one of the first SDN-enabling standards.

With SDN, packet forwarding is flow oriented, meaning both origin and destinations are taken into account, instead of just packet destination, as in traditional networking. The SDN controller manages flow policies for a range of forwarding elements, effectively moving such functions out of the devices. Thus, SDN-capable elements can be dynamically reconfigured over the network accordingly with the needs of network services and applications. For this reason, the controller will have a broader view of the domain, contrasting with the narrow view that an individual forwarding element has in a traditional IP network. **Figure 2** illustrates the flow-rule table of the OpenFlow protocol (one of the most popular SDN protocols).

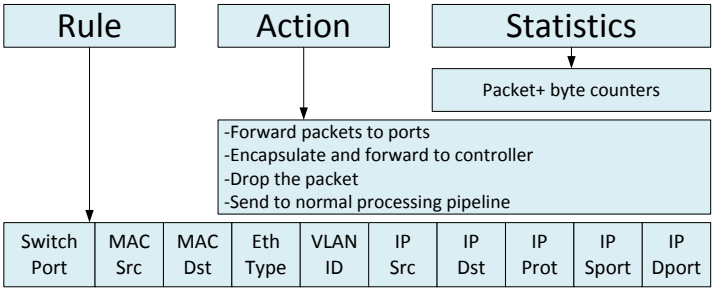

Figure 2: Openflow flow-rule table (adapted from SDX Central 2014)

Leveraging SDN for SCADA/ICS security

SDN allows for increased network flexibility and programmability, in particular for complex scenarios, which benefit from the reduced overhead for management operations such as topology changes for implementing overlay networks. Besides these benefits, SDN can also provide an effective mechanism for security applications (Proença *et al.* 2015). This is due to the fact that a centralized element with a global view of all the network entities (such as devices, flows, and network elements) is able to provide more efficient information-gathering and security-reaction mechanisms, especially when compared with the narrow local view individually provided by each forwarding element in traditional IP networks. For instance, an Openflow controller can provide information useful for online analysis and detection of security issues, as suggested by Braga, Mota, and Passito (2010):

- **Packets per flow:** this counter can be used for slow rate DDoS detection, as such attacks usually rely on the transmission of a reduced number of packets from a large amount of sources;
- **Average bytes per flow:** this can be used to detect small payload sizes, which are frequent in DDoS attack flows, in order to increase the attack efficiency;
- **Average duration per flow:** an SDN flow is deleted from its flow table if left inactive (no packets received) for a period of time, a feature which can be used to detect short flows characteristic of DoS attacks (Sadre, Sperotto & Pras 2012);
- **Percentage of pair-flows:** an asymmetry between flows coming into and out of the network can be an indicator of an ongoing DDoS attack (Kreibich 2005);
- **Number of single-flows:** it is possible that the number of unpaired flows increases dramatically in the beginning of a flood attack. This can be calculated on a per interval basis after subtracting the paired flows from the total;
- **Number of used TCP/IP ports and addresses:** DDoS frequently involve random spoofing of IP and ports, whose rate of increase may reveal ongoing issues.

Moreover, flow-based forwarding can be used to increase the efficiency of a reaction, being used to isolate or divert flows, instead of simply blocking an attack. This is useful to improve existing security techniques—for example, dynamically diverting attackers to honeypot systems as soon they are detected. SDN can also help handling Denial of Service (DoS) and Distributed DoS (DDoS) attacks by improving detection and reaction mechanisms.

Besides the generic security application scenarios, there have been several developments regarding SDN-based security mechanisms for ICS. For instance, Dong *et al.* (2015) propose reinforcing the resilience of SCADA networks used for smart grid applications using a solution relying on three elements (SCADA master, SDN controller, Intrusion Detection System—IDS), which coordinate with each other in order to detect attacks and reconfigure the network so as to mitigate and overcome identified problems. Suggested use cases include the dynamic establishment of routes to transmit control commands only when necessary (to shorten the time window for tampering attempts), automatic rerouting or dropping of suspicious packets to avoid spoofing or flooding attacks from compromised SCADA elements, or implementation of network monitors to deal with delay attacks.

Irfan & Mahmud (2015) propose using SDN for dynamic creation of virtual networks in order to isolate distinct traffic and hosts, and to enable traffic prioritization and secure partitioning. The concept is demonstrated using an SDN-controller proxy to create three isolated networks, which share the same physical infrastructure but have their own SDN controllers. Authors discuss the use of this architecture to improve aspects such as authentication, confidentiality, integrity, non-repudiation, and availability. A similar approach is also suggested by Machii *et al.* (2015) as a way to minimize the attack surface by using SDN to dynamically segregate fixed functional groups within the ICS. A dynamic zone-based approach is also proposed, taking advantage of the information obtained from field devices to estimate the operation phase of the ICS (as each phase—such as start-up, normal operation, or load-change—exhibits different behaviour and communications profiles) and to calculate the optimal zone topology, deploying the needed SDN configuration in runtime. This strategy reduces the time and spatial exposure to attacks (effectively creating a moving target) and also provides the means to isolate compromised devices.

Also related to dynamic configuration techniques, Chavez *et al.* (2015) present a security solution based on network randomization, which also encompasses an IDS with near real time reaction capabilities. This network randomization approach assigns new addresses to network devices in a periodic basis or by request, in order to protect them against attacks that rely on knowledge about the ICS topology (such as static device addresses). The responsible controller application keeps an updated database of all the network specifications (mostly devices and real addresses), generating overlay IP addresses for the same devices and for each flow, which are used to define the OpenFlow rules on flow tables. This way, all the traffic flowing on the network uses 'fake' overlay addresses that are periodically randomized, reducing their useful lifetime and, consequently, the time window available for any attacker to take advantage of that knowledge. The proposed IDS takes advantage of the predictable, auto-similar, traffic patterns of ICS networks for identifying attacks and triggering defence reactions (a network randomization request, which will render useless any ongoing attack using old overlay addresses). Attack detection makes use of machine learning algorithms and mathematical methods, fed and trained using OpenFlow's statistical counters.

Silva *et al.* (2015) also describe a dynamic technique that makes use of SDN to prevent eavesdropping on SCADA networks. The intended goal is to deter attackers from collecting sequential data, which is essential for breaking encryption, identifying patterns, and retrieving useful information from the payload. By taking advantage of redundant network connectivity, a multi-path routing mechanism enables a flow to be transmitted and split over different paths (see **Figure 3**, below) by resorting to an algorithm that calculates the shortest path between two devices, dynamically assigns a cost to each one, and uses an OpenFlow timer (hard timeout) to periodically reinstall new flow rules.

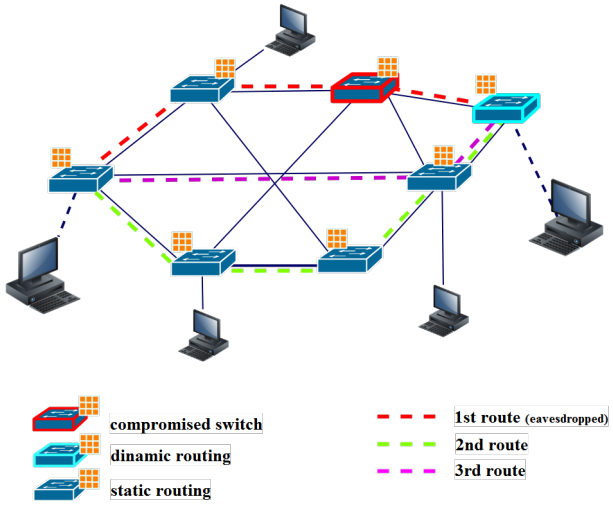

Figure 3: Multi-flow, redundant routing for flow splitting (adapted from Silva *et al.* 2015)

Genge *et al.* (2016) propose two distinct SDN-based techniques to mitigate and block ICS cyber attacks. The first technique (see **Figure 4**, below), designed for single-domain networks, attempts to mitigate DoS attacks by rerouting traffic, using information from the SDN controller. SDN controllers feed an application that continuously monitors the state of the network links and communicates with the controller to issue flow reconfiguration operations. Once an attack is detected (few details are provided about this, though), the corresponding data flows are rerouted, in order to protect the ICS.

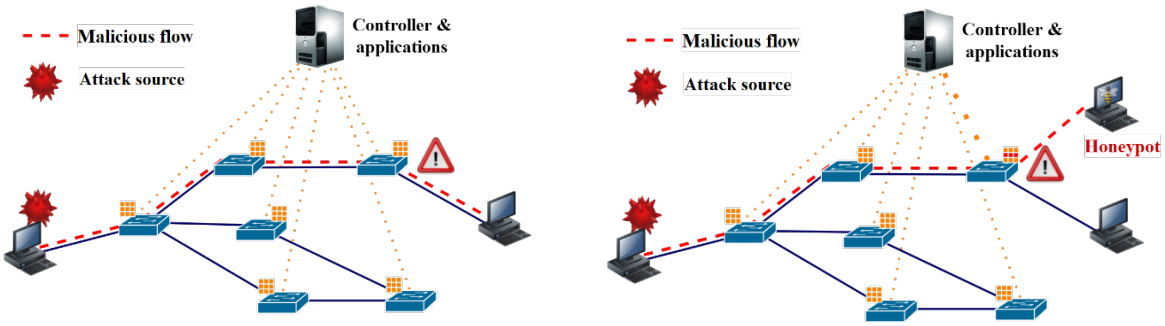

Figure 4: A single-domain SDN-based security solution (adapted from Genge *et al.* 2016)

The second technique (see **Figure 5**) targets multi-domain networks, with the goal of blocking the attack as close as possible to the entry point in the network.

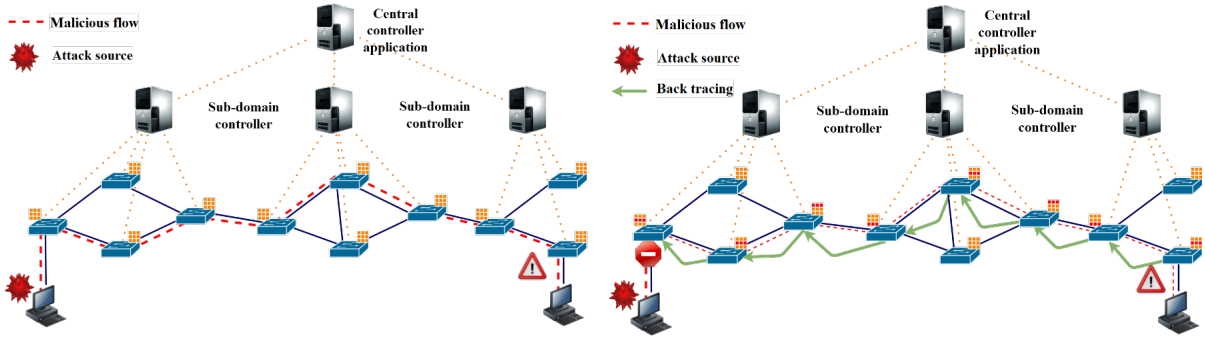

Figure 5: A multiple-domain SDN-based security solution (adapted from Genge *et al.* 2016)

For such a multi-domain network, each domain has its own OpenFlow controller, connected to a centralized security application. This application receives information from the SDN controllers, which have access to a global perspective about the network. Once an attack is detected, the security application will backtrack towards its origin by recursively issuing queries about the related flows to identify the previously paired nodes until the original network entrance point is found.

ICS-specific honeypots and honeynets can also benefit from the introduction of SDN technologies. Honeypots are decoy or dummy targets set up to attract and detect/profile attacks. Exposed to probing and attack, these targets are used to lure and track intruders as they advance (Simões *et al.* 2013), revealing any scouting activities. Traditionally, honeypot systems live in unused address space in the system, waiting for attackers to find them (Spitzner 2003), but their operation can be greatly improved by SDN, which has the possibility of turning them into a more proactive defence.

Using SDN network-flow manipulation capabilities, it is possible to improve honeypot operation and transform it into an active security component by working together with other mechanisms, such as network intrusion detection systems (NIDS). When an unauthorized activity is detected by a NIDS, the SDN controller can divert the anomalous traffic flows to an ICS-specific honeypot, such as the one proposed by Simões *et al.* (2013). The attacker would not be aware of this diversion and would continue the attack. Meanwhile the honeypot will log its activity for forensics analysis. **Figure 6**, below, illustrates an example of this approach.

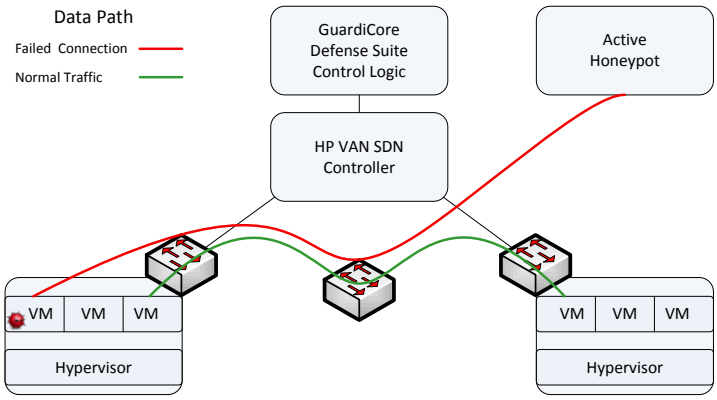

Figure 6: Active honeypot (reproduced from Hewlett-Packard 2014)

Also, Song, Shin, and Choy (2014) suggested using honeynets (networks set up with several honeypot devices) together with SDN technologies to detect scouting procedures and collect profiling information about attackers. This is achieved by providing the attacker with false information from the honeynet, using OpenFlow to detect the scan attacks by inspecting packets coming towards closed or unused ports, or to detect corrupt packets or sessions. After a successful detection, the infringing packet and the subsequent ones in the same flow will be redirected to the honeynet. Despite being a generic proposal, this solution can be easily ported to most ICS infrastructures.

Network Function Virtualization and distributed ICS

NFV is the result of the convergence between telecommunications infrastructures and infrastructure virtualization. As network applications and services scale and evolve (not only in sheer capacity requirements, but also in complexity), they impose an added burden to the supporting telecommunications provider infrastructure, requiring the use of specific network management and traffic policies that cannot be provided by the network. As Chiosi *et al.* (2012) have noted, from this perspective, NFV is a significant development as it enables the creation of flexible and on-demand network services through a service chain-based composition mechanism that uses network functions implemented in VNF (Virtualized Network Functions) components comprising functionality such as NAT, IDS, Firewalls or other service modules implemented as VM appliances.

The NFV vision attempts to decouple network capacity from functionality, by conceiving an end-to-end service as an entity that can be modelled and described by means of network function forwarding graphs (**Figure 7**) involving interconnected VNFs and endpoints (also known as service chaining).

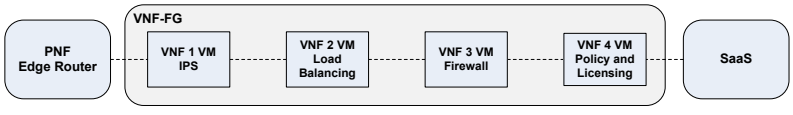

Figure 7: NFV Forwarding Graph example

This approach allows for creation of differentiated end-to-end services that can be provided by the (ordered) combination of elementary VNF or physical functions, chained together by a Forwarding Graph, which models the service flows (see **Figure 8**, below). Furthermore, VNF FGs can be nested to define complex functions. VNFs are implemented in software, being interconnected through the logical links that are part of a virtualized network overlay, which can be implemented using SDN.

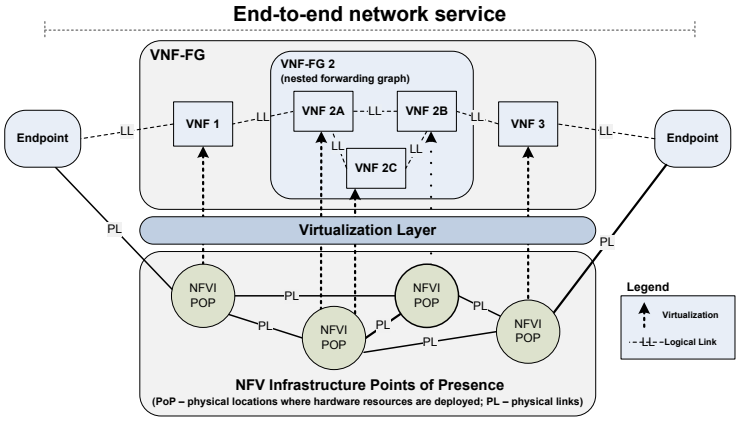

Figure 8: NFV end-to-end service with VNFs (adapted from Ersue 2013)

Eventually, even Physical Network Functions (conventional network devices with close coupled software and hardware that perform network functions) can be involved in a Network Forwarding Graph service chain (the concept of service chain is not exclusive of NFV). A virtualization layer abstracts the physical resources (computing, storage, and networking) on top of which the VNFs are deployed and implemented, with the supporting NFV Infrastructure (NFVI) being spread across different physical locations, called Points of Presence (NFVI PoPs), as shown in **Figure 5**, above.

NFV as an enabler for a new generation of distributed IACS

Use cases such as Internet of Things (IoT), wire-to-water generation, micro generation, smart metering or smart water management constitute a new generation of distributed IACS that can only be supported with the help of a complex distributed software stack, potentially also requiring the involvement of third-parties, such as telecommunications and cloud operator infrastructures—for this reason, the introduction of Network Function Virtualization component appliances, distributed across geographically dispersed infrastructure PoPs, makes entire sense.

As the IACS enters the customer premises, the NFV service abstraction model (services as composition of VNFs) provides an effective way to introduce support components along the service path. For instance, a data collection and analysis VNF can be added to the customer service chain (eventually within a virtual Business Gateway service abstraction) to provide data collection for smart metering scenarios. The same rationale applies for security purposes, as cyber-physical protection (for example, to implement bump-in-the-wire encryption) or security anomaly detection VNFs can be integrated within service chains, also using SDN to create flexible security monitoring and reaction capabilities. Moreover, Distributed IDS (DIDS) components may be consolidated in the form of VNFs optimally deployed in order to reduce service overhead and rationalize resources. For instance, the DIDS components might be deployed in the form of VNFs, either shared among several Business Gateway FGs or used exclusively by a service instance (Cruz *et al.* 2015). Some manufacturers (RAD 2015) (ECI 2015) are starting to propose NFV products for ICS applications that implement this philosophy, incorporating NFV capabilities in access nodes for optical transport or packet switched networks, for hosting firewall, encryption or traffic monitoring VNFs.

NFV is also an enabler for fog computing scenarios. The term ‘fog computing’, frequently also referred as ‘edge computing’, is based on the idea that, rather than hosting and working from a centralized cloud, some parts of the infrastructure may be deployed on network ends,

using virtualized platforms located between end-user devices and the cloud data centres. It attempts to provide better quality of service in terms of delay, power consumption, and reduced data traffic over the Internet, among other benefits. Fog computing tries to address the need to process large data streams in real time while working within the limits of available bandwidth, by placing some of transactions and resources at the edge of the cloud, thus improving the efficiency of the infrastructure by offloading processing tasks before passing them to the cloud.

The NFV paradigm is naturally compatible with fundamental premises for implementation of fog-computing distributed topologies. As such, it is envisioned that distributed awareness and IACS cyber-security detection capabilities will take advantage of the NFV paradigm to support their underlying deployment model, departing from the conventional, self-contained model and moving towards an architecture capable of keeping up with the geographically dispersed nature of IoT IACS. Also, the VNF deployment criteria may consider the availability of specific capabilities (such as raw processing capacity) in a specific NFVI POP. For instance, per-subscriber security-event processing components may be hosted in a different NFVI POP from the one(s) hosting other VNFs for the DIDS service.

Real-time Hypervisors + SDN = Towards a Virtualized PLC

Born in the mainframe era, Virtual Machine Monitors (also called Hypervisors) have ultimately evolved towards being supported in open, Commercial Off-The-Shelf (COTS) hardware, bringing a significant improvement for the ICT ecosystem, allowing for co-hosting of several VMs within a host machine, sharing resources, and providing a managed execution environment. Specifically, type-1 (bare metal) hypervisors have become popular in large-scale virtualization scenarios such as data centres, bringing several benefits in terms of resource consolidation, business continuity, scalability, management, and security.

However, most type-1 hypervisors are optimized for ICT loads, and, thus, are unsuitable for several ICS application use cases, mostly due to the overhead of the mediation and translation mechanisms abstracting the host hardware from the VM. This situation gradually began to change, as some operators started virtualizing hosts with services deployed on general-purpose OS, such as SCADA Master Stations (MS), Human-Machine Interfaces (HMI) or Historian Database servers (HDB), using conventional type-1 hypervisors. This was possible due to developments that allowed such hypervisors to benefit from hardware-assisted memory management and I/O mechanisms to implement robust resource affinity and reservation (such as VT-d and PCI SRV-IO; see Garcia-Valls, Cucinotta & Lu 2014), thus, providing performance guarantees while avoiding the effect of resource overprovisioning. Also, real-time clock integrity issues, one of the main concerns in hypervisor environments, were mostly solved using para-virtualized interfaces (KVM 2015) and/or adequate clock synchronization policies.

Other ICS elements, such as process control devices, can also potentially benefit from virtualization technologies. For instance, (Cahn *et al.* 2013) proposed the virtualization of Intelligent Electronic Devices (IEDs) used to collect information from sensors and power equipment, with the purpose of optimizing the maintenance and cost overheads, while increasing reliability. The same rationale could be applied to Programmable Logic Controller (PLC) devices, which constitute the focus of this section.

PLCs are pervasive components in ICS, such as SCADA systems, being designed to control industrial processes autonomously or as part of a distributed-control system topology. While

the success of the PLC may be explained by its robustness and reliability, it is one of the most enduring legacies in modern ICS, having evolved very little over the last years. Modern PLCs are the outcome of an evolutionary process that started with the first generation of relay-based devices, progressively incorporating technologies such as microprocessors and microcontrollers, Real-Time Operating Systems (RTOS) and communications capabilities ranging from serial point-to-point or bus topologies to Ethernet and TCP/IP. Although modern PLCs are often embedded devices running Real-Time Operating Systems (RTOS), equipped with System-on-Chip or CPUs (PowerPC, x86 or ARM) based on commodity Instruction Set Architectures (ISA), their virtualization was not deemed feasible until recently, due to the lack of specific hardware, software, and infrastructure support.

Towards the virtual PLC

PLCs are designed for reduced and deterministic latency, operating under strict timing constraints that are dependent on factors such as the end-to-end and event response latencies across components on interconnected buses, or signal and message propagation delays. These requirements are incompatible with the use of several virtualization technologies, such as conventional type-1 hypervisors, due to overhead issues and the lack of support for real-time payloads.

However, recent developments, such as the implementation of low-latency deterministic network connectivity for converged Ethernet and the availability of real-time hypervisors, have made it possible to virtualize components of the PLC architecture. The vPLC architecture described by Cruz, Simões & Monteiro (2016) takes advantage of these capabilities by decoupling the PLC execution environment from I/O modules using an SDN-enabled Ethernet fabric to provide connectivity to the I/O subsystem (**Figure 9**, below). This architecture departs from the SoftPLC concept, as proposed by products such as (Codesys) or (ISaGRAF), by adopting an approach in line with (Intel 2013) and (IntervalZero 2010), with the added benefit of a convergent fabric scenario with SDN capabilities.

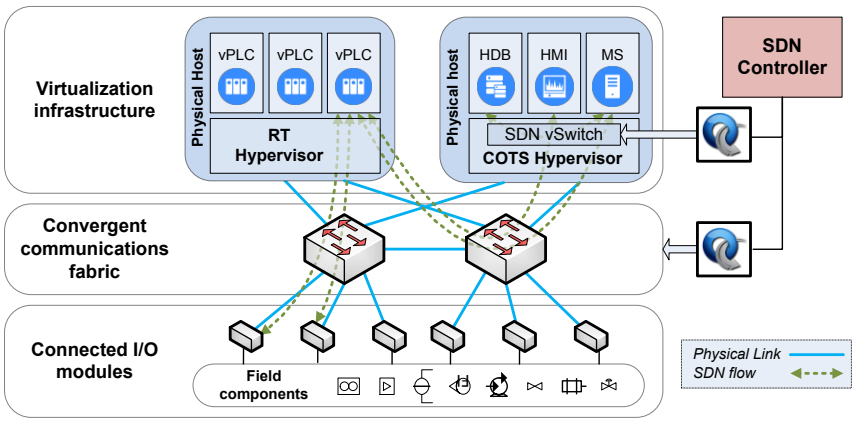

Figure 9: The vPLC architecture

In the vPLC, the PLC I/O bus is replaced by high-speed networking capabilities, with SDN allowing for the creation of flexible virtual channels on the I/O fabric, accommodating the connectivity flows between the vPLC instances and the I/O modules (such as sensor interfaces or motion controllers), and providing traffic isolation. Moreover, such I/O modules can be built with reduced complexity, thanks to recent progress in terms of Field-Programmable Gate Arrays (FPGA) and Application Specific Integrated Circuit (ASIC) technology. SDN reconfiguration is managed by means of an SDN controller, via a High-

Availability (HA) server (not depicted in the figure), which interacts with its northbound interface. The HA server continuously monitors the SDN switch statistics and path reachability, triggering reconfiguration procedures in case of performance degradation or failure.

This decentralized model shares similarities with remote or distributed I/O PLC topologies, with networked I/O modules acting as extensions of the PLC rack. This architecture shares similarities with the Converged Plantwide Ethernet (CWpE) (Didier *et al.* 2011) proposal, or even critical avionics systems, which replace legacy interconnects with Ethernet-based technologies, such as Avionics Full-Duplex Switched Ethernet (AFDX) (Fuchs 2012).

Advances in cut-through switching, together with Remote Direct Memory Access techniques (RDMA), particularly in converged Ethernet scenarios, have allowed for port-to-port latencies of the order of the hundredths of nanoseconds in 10G Ethernet switch fabrics and application latencies in the order of microseconds (Beck & Kagan 2011). Additionally, resources such as Intel's Data Plane Development Kit (DPDK) (Zhang *et al.* 2014) allow for the implementation of low-latency, high-throughput packet processing mechanisms that bypass kernels, thus, bringing the network stack into user space and enabling adapters to perform Direct Memory Access operations to application memory. This enables satisfying requirements for single-digit microsecond jitter and restricted determinism, allowing for bare-metal performance on commodity server hardware. On top of this, proposals such as the 802.1Qbv Time Sensitive Networking (IEEE TSN) standard provide compliance with real-time requirements in the microsecond range on conventional Ethernet.

As for computing resources, there are two factors that must be considered. First, modern x86 or ARM processors have become capable of replacing microcontrollers in standalone PLC applications (Kean 2010) because of improvements in terms of raw performance, low latency I/O mechanisms, or the availability of ISA extensions suitable for Digital Signal Processing tasks. Second, the availability of real-time static partitioning hypervisors, such as Jailhouse (Siemens), Xtratum (Crespo, Ripoll & Masmano 2010), X-Hyp (X-HYP) or PikeOS (Baumann *et al.* 2011), enables hosting RTOS guest VMs for real-time workloads. Some hypervisors, such as Xtratum and PikeOS, even replicate the ARINC 653 (Fuchs 2012) partitioning model for safety-critical avionics RTOS, with a Multiple Independent Levels of Security/Safety (MILS) (Alves-Foss *et al.* 2006) architecture.

The benefits of this approach are manifold. The price tag for entry-level PLCs is comparable to a COTS server that can host several vPLC instances, being kept out of the factory floor or industrial environment. Distributed I/O on converged Ethernet also provides cost-effective performance and reliability benefits, as communications between different vPLC instances can take place across the convergent fabric or even locally, if co-located on the same host, with SDN allowing for flexible creation of communications channels for differentiated requirements. Moreover, I/O modules—the components with highest failure rate in PLCs—can be easily and quickly replaced in case of failure.

Particularly, the potential advantages of the vPLC in terms of reliability, safety, and security are considerable, as it can take advantage of datacentre-like redundant power, computing, and communications resources. Other benefits are also envisioned, namely:

- Hypervisors allow for migration of virtualized ICS components, as well as instance cloning for pre-deployment tests;

- PLC watchdogs and system-level debugging and tracing mechanisms can be implemented at the hypervisor level, which is able to oversee and control the vPLC partition behavior;
- vPLCs benefit from partitioning isolation, with VMs being easy to restore in a fresh state in case of tampering or other malicious activity;
- SDN-managed isolated I/O paths ease the implementation of flexible, on-demand protection mechanisms at the I/O level, thereby paving the way for the introduction of NFV components at the ICS level.

Overall, these benefits constitute strong arguments in support of the vPLC proposal. Moreover, most of them suggest that the vPLC could be feasible even for a single instance per device, using Industrial-grade Single Board Computers, instead of COTS servers.

Conclusion

This paper discusses the implications of the progressive introduction of virtualization technologies in ICS, with a special focus on security and reliability aspects. The virtualization of both network and computing virtualization was analysed from an ICS-centric standpoint, covering recent developments as well as proposing new use cases and approaches to improve network and systems security.

Starting with an overview of network virtualization technologies, such as SDN and NFV and their application within ICS and distributed IACS, the paper next addressed the issue of using hypervisor technologies for real-time workloads. In this latter perspective, a virtual PLC (vPLC) architecture was discussed, which transcends the simple virtualization of the PLC device, constituting an integrated approach in which the device merges with the infrastructure in a seamless way. The vPLC takes advantage of network and computing virtualization technologies to propose a converged approach for plan-wide consolidation of the ICS infrastructure, with performance, cost, and security benefits. This proposal is presently under development by a team that includes the authors of this paper.

Acknowledgements

This work was partially funded by the ATENA H2020 Project (H2020-DS-2015-1 Project 700581).

References

Alves-Foss, J, Harrison, W, Oman, P, & Taylor, C 2006, 'The MILS architecture for high assurance embedded systems',. *International Journal of Embedded Systems*, vol. 2, no. 3-4, pp. 239–47.

Baumann, C, Bormer, T, Blasum, H, & Tverdyshev, S 2011, 'Providing memory separation in a microkernel by code level verification', *Proceedings of the 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, pp. 25–32.

Beck, M & Kagan, M 2011, 'Performance evaluation of the RDMA over Ethernet standard in enterprise data center infrastructure', *Proceeding of the 3rd Workshop on Data Center-Convergent and Virtual Ethernet Switching,* pp. 44-62.

Braga, R, Mota, E & Passito, A 2010, 'Lightweight DDoS flooding attack detection using

NOX/OpenFlow', *Proceedings of the 2010 IEEE 35th Conference on Local Computer Networks (LCN)*, pp. 408-15.

Cahn, A, Hoyos, J, Hulse, M & Keller, E 2013, 'Software-defined energy communication networks: from substation automation to future smart grids', *Proceedings of IEEE SmartGridComm 2013 Symposium: Smart Grid Services and Management Models*, pp. 558-563.

Chavez, A, Hamlet, J, Lee, E, Martin, M & Stout, W 2015, *Network randomization and dynamic defense for critical infrastructure systems*', Sandia National Laboratories, Albuquerque, New Mexico and Livermore, California, USA.

Chiosi, M, Clarke, D, Wilis, P, Reid, A, Feger, J, Bugenhagen, M, Khan, W, Fargano, M & Others 2012, *Network functions virtualization—an introduction, benefits, enablers, challenges & call for action, issue* ', ETSI white paper, October 2012, viewed 12 July 2016, <http://portal.etsi.org/NFV/NFV_White_Paper.pdf>.

Codesys 2016, CODESYS Control RTE: Real-time SoftPLC under Windows, viewed 12 July 2016, <https://www.codesys.com/products/codesys-runtime/control-rte.html>.

Crespo, A, Ripoll, I, & Masmano, M 2010, 'Partitioned embedded architecture based on hypervisor: the XtratuM approach', *Proceedings of the European Dependable Computing Conference (EDCC)*, pp. 67-72 .

Cruz, T, Simões, P, Monteiro, E, Bastos, F, & Laranjeira, A 2015, 'Cooperative security management for broadband network environments', Wiley Security and Communication Networks, vol. 8, no. 18, pp. 3953-77.

Cruz, T, Queiroz, R, Simões, P & Monteiro, E 2016, 'Security implications of SCADA ICS virtualization: survey and future trends', *Proceedings of 15th European Conference on Cyber Warfare and Security* (ECCWS), July, pp. 74-83.

Cruz, T, Simões, P & Monteiro, E 2016, 'Virtualizing Programmable Logic Controllers: towards a convergent approach', IEEE Embedded Systems Letters, September 2016, DOI: 10.1109/LES.2016.2608418.

Didier, P, Macias, F, Harstad, J, Antholine & R, Johnston, S 2011, 'Converged Plantwide Ethernet (CPwE) Design and Implementation Guide', , Cisco Inc. and Rockwell Automation white paper.

Dong X, Lin, H, Tan, R, Iyer, R & Kalbarczyk, Z 2015, 'Software-Defined Networking for smart grid resilience: opportunities and challenges', *Proceedings. of the 1st ACM Cyber-Physical System Security Workshop* (CPSS'15), Singapore, 2015, pp. 61-68.

ECI Telecom 2015, *LightSEC NFV-based cyber security solution for utilities*, , viewed 12 July 2016, <http://www.ecitele.com/media/1225/eci_lightsec_nfv_brochure-utilities.pdf>.

Ellanti, M, Scott Gorshe, S, Raman, LG & Grover, WD 2005, *Next generation transport*

networks: data, management, and control planes, Springer-Verlag, US, DOI 10.1007/b104435.

Ersue, M 2013, 'ETSI NFV Management and orchestration - an overview', Presentation at the IETF #88 meeting, Vancouver, Canada, 3-8 November 2013, viewed 12 July 2016, <http://www.ietf.org/ proceedings/88/slides/slides-88-opsawg-6.pdf>.

Fuchs, C 2012, 'The Evolution of avionics networks from ARINC 429 to AFDX', *Proceedings of Innovative Internet Technologies and Mobile Communications and Aerospace Networks*, vol. 65, pp. 65–76.

Galup, SD, Dattero, R, Quan, JJ & Conger, S 2009, 'An overview of IT service management', *Communications of the ACM*, vol. 52, no. 5, pp. 124-27, DOI: 10.1145/1506409.1506439.

García-Valls, M, Cucinotta, T, & Lu, C 2014, 'Challenges in real-time virtualization and predictable cloud computing', *Journal of Systems Architecture*, vol. 60, no. 9, pp. 726-40.

Genge, B, Haller, P, Beres, A, Sándor, H & Kiss, I 2016, Using Software-Defined Networking to mitigate cyberattacks', *Securing Cyber-Physical Systems*, CRC Press, pp. 305-29, ISBN:9781498700986, Boca Raton, FL, USA.

Greene, K 2009, 'Tech Review 10, breakthrough technologies: Software-Defined Networking', *MIT Technology Review*, viewed 12 July 2016, <http://www2.technologyreview.com/news/412194/tr10-software-defined-networking>.

Hewlett-Packard 2014, *Data center security redefined*, viewed 12 July 2016, <http://h20195.www2.hp.com/v2/GetPDF.aspx%2F4AA5-1629ENW.pdf>.

IEEE, Time-Sensitive Networking Task Group, viewed 12 July 2016, available from <http://www.ieee802.org/1/ pages/tsn.html>.

Igure, V, Laughter, S & Williams R 2006, 'Security issues in SCADA networks', *Computers & Security*, vol. 25, no. 7, pp. 498-506.

Intel Corporation 2013, *Reducing cost and complexity with industrial system consolidation*, viewed 12 July 2016, <http://www.intel.com/content/www/ us/en/industrial-automation/reducing-cost-complexity-industrial>.

IntervalZero 2010, 'A soft-control architecture: breakthrough in hard real-time design for complex systems', viewed 12 July 2016, <http://intervalzero.com/assets/wp_softControl.pdf>.

Irfan, N & Mahmud, A 2015, 'A novel secure SDN/LTE-based Architecture for Smart Grid Security', *Proceedings of the 2015 IEEE International Conference on Computer and Information Technology*, pp. 762-769.

ISA-99.00.01 2007, 'Security for industrial automation and control systems, part 1: terminology, concepts, and models', American National Standard.

ISaGRAF. ISaGRAF overview, viewed 12 July 2016, <http://www.isagraf.com>.

Kang, DJ, Lee, JJ, Kim, BH & Hur, D 2011, 'Proposal strategies of key management for data encryption in SCADA network of electric power systems', International Journal of Electrical Power & Energy Systems, vol. 33, no. 9, pp. 1521-1526.

Kean, L 2010, 'Microcontroller to Intel architecture conversion: PLC using Intel atom processor', Intel Corp. technical note, viewed 12 July 2016, <http://www.intel.com/content/www/us/en/embedded/training/microcontroller-ia-conversion-paper.html>.

Kreibich, C & Warfield, A 2005, 'Using packet symmetry to curtail malicious traffic', in proc. of 4th ACM Workshop on Hot Topics in Networks (Hotnets-IV), November 2005, USA.

Kreutz, D, Ramos, F, Verissimo, P, Rothenberg, C, Azodolmolky, S & Uhlig, S 2014, 'Software Defined Networking: a comprehensive survey', *Proceedings of the IEEE*, vol. 103, no. 1, pp.14-76.

Krutz, RL 2006, *Securing SCADA systems*, Wiley Publishing, Inc., Hoboken, NJ, U.S.A.

KVM Project 2015, KVM PVclock, viewed 12 July 2016, available from <http://www.linux-kvm.org/page/KVMClock>.

O'Murchu, L & Falliere, N 2011, 'W32.Stuxnet dossier', Symantec white paper, February 2011.

Machii, W, Kato, I, Koike, M, Matta, M, Aoyama, T, Naruoka, H, Koshima I & Hashimoto, Y 2015, 'Dynamic zoning based on situational activities for ICS security', Proceedings of the 10th Asian Control Conference (ASCC), pp. 1-5.

ONF 2012, *OpenFlow switch specification, version 1.3.0 (Wire Protocol 0x04)*, Open Networking Foundation, viewed 12 July 2016, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications /openflow/openflow-spec-v1.3.0.pdf>

Proença, J, Cruz, T, Monteiro, E & Simões, P 2015, 'How to use software-defined networking to improve security–a survey', *Proceedings of the 14th European Conference on Cyber Warfare and Security*, pp. 220-228.

RAD Data Communications Ltd. 2015, *Megaplex-4 D-NFV Virtualization Module*, viewed 12 July 2016, <http://www.rad.com/Media/34173_D-NFV.pdf>.

Sadre, R, Sperotto A & Pras, A 2012, 'The effects of DDoS attacks on flow monitoring applications', in proc. of 2012 IEEE Network Operations and Management Symposium, Maui, HI, 2012, pp. 269-277.

Schudel, G & Smith, D 2007, *Router security strategies: securing ip network traffic planes*', Cisco Press, Indianapolis, IN, U.S.A., ISBN: 1-58705-336-5.

SDX Central 2014, *What is OpenFlow?* , viewed 12 July 2016, <https://www.sdxcentral.com/resources/sdn/what-is-openflow>.

Siemens AG n.d., *Jailhouse Partitioning Hypervisor*, viewed 12 July 2016, <https://github.com/siemens/jailhouse>.

Silva, E, Knob, L, Wickboldt, J, Gaspary, L, Granville, L & Schaeffer-Filho, A 2015, 'Capitalizing on SDN-based SCADA systems: an anti-eavesdropping case-study', *Proceedings of the IFIP International Symposium on Integrated Network Management (IM 2015)*, pp. 65-173.

Simões, P, Cruz, T, Proença, J & Monteiro, E 2013, 'On the use of Honeypots for Detecting Cyber Attacks on Industrial Control Networks', *Proceedings of the 12th European Conference on Information Warfare and Security (ECIW 2013)*, pp. 263-270.

Song, Y, Shin, S & Choi, Y 2014, 'Network Iron Curtain: Hide Enterprise Networks with OpenFlow'. Y Kim, H Lee, & A Perrig (eds.), *Information security applications, lecture notes in computer science*. Springer International Publishing, New York, NY, USA, pp. 218-30.

Spitzner, L 2003, *Honeypots: tracking hackers*, Addison-Wesley, Boston MA, USA, ISBN:0321108957.

Triangle MicroWorks, Inc. 2002, *DNP3 overview*, viewed 12 July 2016, <http://www.trianglemicroworks.com/documents/DNP3_Overview.pdf >.

X-HYP Project, viewed 12 July 2016, <from: http://x-hyp.org>.

Yeganeh, S, Tootoonchian, A & Ganjali, Y 2013, 'On scalability of software-defined networking', *Communications Magazine*, IEEE, vol. 51, no. 2, pp.136-41.

Zhang, W, Wood, T, Ramakrishnan, K, and Hwang, J 2014, 'Smartswitch: blurring the line between network infrastructure and cloud applications', *Proceedings of the 6th USENIX Workshop on Hot Topics in Cloud Computing*.

Zhu, B, Joseph, A & Sastry, S 2011, 'A taxonomy of Cyber Attacks on SCADA Systems', *Proceedings. of the 2011 International. Conference. on Internet of Things and 4th International. Conference. on Cyber, Physical and Social Computing* (ITHINGSCPSCOM'11), pp. 380-388.