Encryption in ICS networks: a Blessing or a Curse?

Davide Fauri[1], Bart de Wijs[2], Jerry den Hartog[1], Elisa Costante[3], Emmanuele Zambon[3] and Sandro Etalle[1,3]

[1]Technische Universiteit Eindhoven {d.fauri,j.d.hartog,s.etalle}@tue.nl

[2]ABB Group bart.wijs@nl.abb.com

[3]SecurityMatters elisa.costante@secmatters.com emmanuele.zambon@secmatters.com

***Abstract*— Nowadays, the internal network communication of Industrial Control Systems (ICS) usually takes place in unencrypted form. This, however, seems to be bound to change in the future: as we write, encryption of network traffic is seriously being considered as a standard for future ICS. In this paper we take a critical look at the pro's and con's of traffic encryption in ICS. We come to the conclusion that encrypting this kind of network traffic may actually result in a reduction of the security and overall safety. As such, sensible versus non-sensible use of encryption needs to be carefully considered both in developing ICS standards and systems.**

I. Introduction

SCADA (Supervisory Control and Data Acquisition) systems and DCS (Distributed Control Systems) form an important subset of ICS (Industrial Control Systems), overseeing complex physical processes in industrial and critical infrastructures which usually span over a large geographic area (e.g. a pipeline, an electrical grid). Over the last decades, ICS have evolved from largely isolated systems to largely interconnected ones, boosting efficiency but opening up the possibility of cyberattacks; indeed, in the last decade, we have witnessed a number of attacks on ICS [5], [25], [10], [7], [23], [4], [12], [30], [31], [15], [2].

The response from the ICS community has been to increase the attention to the security mechanisms already in place, and to look for new ways to defend against malicious entities. One of the proposed mechanisms to secure ICS is to encrypt communications transmitted over SCADA networks. A few proposals are "on the table" and, at the time of writing this article, there is a committee discussing a possible standardization for the use of encryption on ICS networks.

It is well-known that security always comes at a cost, which is not only monetary, but also in terms of e.g. usability of the system [27]. It is therefore important to evaluate whether a solution is actually worth its costs. To make such an evaluation one has to take into due consideration the attacker model at hand, the possible attacker model in the future, and the business model of the stakeholders in the ICS.

This paper aims at contributing to the discussion on the pro's and con's of network encryption for ICS by providing a basis for analysing the costs and the benefits of such a solution. We determine key threats by considering recent reported ICS attacks. As the business model of the specific target ICS will also influence the discussion, the reasoning and the conclusions of this work have to be "instantiated" intelligently to the various application fields. Yet there are some generally applicable conclusions we believe apply to ICS architectures in general.

The first conclusion is that, in most cases, introducing encryption (in the ICS internal network) does not yield extra security. None of the attacks we considered would have been blocked or made more difficult by the addition of encryption. Encryption aims at mitigating confidentiality leaks "on the wire", while the witnessed attacks target end-points. Also, in many of the attacks, confidentiality is not the security goal being breached. We know of no record of an "attack on the wire" occurring in practice, while many damaging hypothetical attacks may be mitigated by authentication checks rather than encryption.

The second conclusion is that encryption can actually have negative consequences for security. For instance, many attacks can be detected with state-of-the-art Network Intrusion Detection Systems (NIDS), provided that the NIDS has access to the communication contents. Of course, one can implement encryption with appropriate "taps" for intrusion detection, but this adds to the cost of the solution.

The third and last conclusion is that encryption *can* considerably raise the costs of troubleshooting and recovery. For instance, problems (e.g., communication troubles, retransmissions, failing devices, etc.) can be identified (much) more quickly and easily in an unencrypted network than in an encrypted one.

We do not advocate completely ruling out encryption of ICS network traffic: in some cases it makes a lot of sense (for instance, "long-haul" connections over untrusted networks, and in systems operating in an adversarial environment). Instead, we advocate healthy reasoning on what encryption is actually good for, and what are its costs, particularly in terms of the loss of safety and security it may actually *introduce*. Note also that in most situations in the ICS world, one only needs to achieve authentication and integrity of the communication, and this can be done without full-fledge encryption (the latter being needed only to guarantee confidentiality.)

In the remainder of this paper, we first establish the setting in Sections II-IV by providing a general description of SCADA systems, their key security requirements related to encryption and the main cryptographic protocols being considered for use as standards for SCADA systems. Next, we determine key threats by looking at recent attacks on SCADA systems in Section V. We then support each of the three conclusions above in Sections VI-VIII before providing

conclusions in Section IX.

II. SCADA SYSTEMS OVERVIEW

In this section we introduce the basics, the architecture and the communication strategies of SCADA systems as a basis for the security discussion in the following sections. Both ICS and SCADA systems monitor and control physical processes. A key feature of SCADA systems is that they operate over multiple geographical locations and, as such, their communication networks need to span over large distances.

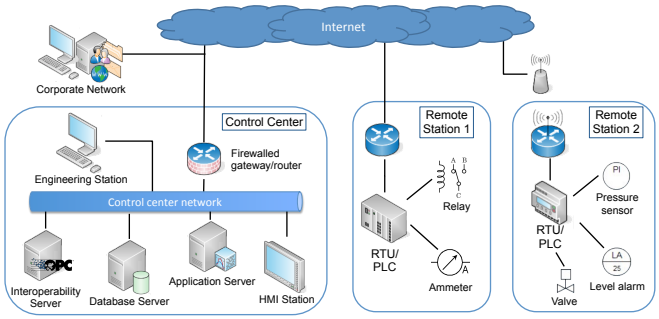

Fig. 1. Simplified architecture of a SCADA system[1]

Figure 1 presents a simplified model of an industrial control system connected through a SCADA network which is sufficient for our purposes. Several geographically distributed remote stations are interconnected with a control center. This could be through a dedicated link or via the Internet.

Each of the stations deals with a different part of a physical process, gathering data through sensors (e.g. the pressure sensor in Remote Station 2), and/or controlling the process through actuators (e.g. the valve at the same station). These end devices are monitored and controlled over a local network by Programmable Logic Controllers (PLC) and Remote Terminal Units (RTU). These are in turn interconnected to each other, possibly in hierarchical master/slave architectures or across remote stations, in order to coordinate the monitoring of the process.

Often industrial systems also have a dedicated control center (CC) to govern the entire process. A typical CC consist of different components, such as SCADA application servers to monitor and control the process, Human-Machine Interfaces (HMI) for operators to interact with the SCADA software, database servers with historical records, or interoperability servers (using standards such as IEC 61850 or OPC-UA, defined in IEC 62541 [6]) for interconnecting SCADA software and hardware devices from different vendors. The CC is usually physically separated from other parts of the system, and relies on a gateway/router to communicate with the remote stations.

Originally, the connection between the CC and the remote stations was done through narrowband radio, dedicated wired links or even satellite systems. The need for integration of services (i.e. firmware update, remote access) has removed the tight separation between SCADA and business networks; and to standardize communications over all these different physical media, SCADA networks are moving to using IP-based networking [20]. For backwards compatibility, messages are repackaged into a TCP/IP wrapper allowing reuse of message formats and existings protocols, such as Modbus. A router/gateway at each remote station serves as interface between IP-based networks on the outside and the fieldbus protocol-based SCADA networks on the facility floor.

The communication between the control center and devices within remote stations can be categorized into four types [33], namely: data acquisition requests, firmware upload, control functions and broadcast messages. These different types of messaging are usually implemented through a request/response model with clear text messages, following a device vendor proprietary communication protocol.

With these main ICS/SCADA network components in place, we next look at the security needs of such systems.

III. SECURITY PROPERTIES AND ENCRYPTION

Encryption is often seen as a method to improve the security of a system. However, to really evaluate the security of a system we first need to know its security requirements.

Capturing security requirements (for ICS). The security requirements for an ICS can be expressed using the classic C.I.A. triad of confidentiality, integrity, and availability, along with authenticity. These are useful to capture the security requirements for any information system. However, priorities of different security requirements in an ICS are inherently distinct from those of a typical IT environment.

In ICS, timely process execution *availability* is the absolute priority, especially for critical infrastructure or a core process of the production line [36]. Process availability is achieved through the sub-requirements of network availability and data correctness, which are also essential to ensure continuous monitoring of faults, anomalies, and potential threats [11]. Correctness of data sent over an untrusted network requires message *authenticity*, which is a combination of *source authentication*, i.e. establishing the identity or role of the sender of a message, and *message integrity*, i.e. assuring data has not been altered during transmission. If the data is valuable, private, or otherwise confidential, we also need message *confidentiality*.

Traditionally, SCADA networks were built on the assumption that only trusted components and entities would be able to connect to them. Thus there were no confidentiality concerns, and integrity checks against faults were sufficient to also achieve messages authenticity. However, nowadays SCADA networks are more accessible and may utilize untrusted networks such as the internet, requiring enforcement and validation of message authenticity, and data confidentiality.

Achieving security requirements. Different cryptographic techniques may satisfy the requirements mentioned above by concealing and/or validating communications. A common interpretation, which we follow in this paper, of the term *encryption* (of traffic) is that of obfuscating the content

[1]Icons source: www.vrt.com.au/downloads/vrt-network-equipment

of messages, i.e. enciphering messages for confidentiality. Encrypted messages can then be read only by parties in possession of the appropriate decryption key: typically, this restricts visibility to just the endpoints of the connection.

Cryptographic techniques can be used to authenticate a party and its messages, for example through the use of public key cryptography with keys validated by digital certificates issued by trusted third parties. We will refer to any cryptographic technique and key/certificate management strategy to achieve authenticity as an *authentication* scheme.

Note that, depending on the cipher and the way it is applied, encryption (i.e. enciphering for confidentiality) may also help to check the integrity and establish the authenticity of messages; encryption and authentication may be achieved by the same cryptographic operation. However, as we are trying to clarify the reasons for using specific techniques, we will still address them as separate requirements.

IV. Encryption protocols for SCADA

ICS standards suggest several protocols to achieve encryption. For example IEC 62351 [8], for power systems infrastructure, recommends end-to-end protocol TLS and point-to-point protocol IPsec; while OPC-UA, for industrial automation systems, refers to end-to-end protocol WS-Security. Here we discuss the protocols recommended by IEC 62351 and use them as examples during the discussion. However, the conclusions that we draw in this paper are not restricted to just these two schemes or the field of power systems: since we discuss in terms of general security properties, the main reasoning remains applicable to the whole field of securing SCADA networks.

According to IEC 62351, Transport Layer Security (TLS) is to be added to the most common TCP/IP industrial protocols such as MMS, DNP3, and IEC 60870-5-104; moreover, the standard discusses the applicability of well-proven standards from the IT domain, such as IPsec.

TLS. TLS creates sessions that provide entity authentication, payload secrecy and message integrity. It accomplishes this by setting up secure sessions using asymmetric public/private keys and digital certificates issued by trusted third-party entities known as Certificate Authorities (CA). A Message Authentication Code (MAC) is appended to each message in a TLS connection to validate a packet's integrity and avoid replay attacks. The MAC is generated from the message's data payload and a shared secret key. Setting up a session consists of two round trips: the first authenticates the server to the client, who validates the server's digital certificate signature against a list of trusted CA in the client's possession. Client authentication is usually left to the application layer, see e.g. IEC 62351 and OPC-UA. The second round trip completes the handshake by negotiating which cryptographic protocol to use, along with a corresponding unique symmetric 'session' key. This key is used to encrypt the content of the messages exchanged during the session: since TLS works at the transport layer, it does not encrypt the routing information on the lower network layer. An external observer that intercepts a TLS secured datagram is limited in the amount of information that he can extract from it: only the endpoints of the communications, along with the type of encryption and approximate size of the data are revealed.

IPsec. The IPsec protocol [19] concerns the network layer and can be implemented in legacy networks as a bump-in-the-wire, i.e. without altering the endpoints.

An IPsec connection is initiated in two phases, according to the Internet Key Exchange (IKE) protocol: Phase 1 has the purpose of generating the shared secret keying material to establish a secure authenticated channel between two peers. Using this channel, Phase 2 negotiates the IPsec security policies to be applied to the data flow, and encrypts the data flow using the keys from Phase 1. After the connection is over, those keys are discarded. To authenticate peers, IPsec uses pre-shared keys, or digital certificate signed by a CA.

IPsec provides two extension protocols: Authentication Header (AH)[17] and Encapsulating Security Payload (ESP)[18]. AH offers data integrity and source authentication for both IP header and payload. As the packet's content is not encrypted, it can still be inspected by a firewall or an IDS. ESP offers data integrity, source authentication, and encryption, and is therefore more widely used in practice; note, however, that the ESP protocol is only applied to the payload and not to the IP header. IPsec is used in one of two modes: tunnel or transport, of which tunnel mode is recommended for establishing secure site-to-site communications from an untrusted network to the control network in SCADA systems [29], [34]. In either mode, the payload is encrypted (using ESP) or authenticated (using AH). In tunnel mode headers are also protected, as the source endpoint encrypts (or authenticates) the entire packet and then encapsulates it in another IP packet. The receiving gateway will then perform the unpacking, decryption (or authentication check) and internal routing necessary to transmit the packet to the final destination device on the trusted network. Tunnel mode can be gateway-to-gateway or host-to-gateway; in either case, the authentication and confidentiality provided by IPsec stop at the receiving gateway and are not fully end-to-end.

V. Attacks on SCADA systems

When checking whether a given approach indeed achieves a security goal, one needs to consider the type of attacks against which they are supposed to defend. To create a broad and representative overview on the current threats to SCADA systems, we have listed (see the first column of Table I) confirmed attacks on SCADA systems from the RISI incident database [1] and recent Verizon data breach digests [30], [31]. Note that we restrict our attention to 'real' attacks: e.g. [36] gives a list of vulnerabilities and potential misuses, some preventable by encryption, but they do not match what is seen in practice. We describe three successful attacks in more detail, namely:

- Stuxnet, causing physical damage to equipment;
- Dragonfly, stealing intellectual property data;
- BlackEnergy, disrupting a wide public infrastructure.

Stuxnet. The Stuxnet malware attack was conducted in 2010, targeting Iranian nuclear enrichment facilities [23]. Stuxnet operated in three stages [14].

In the first stage, the initial infection was likely conducted via an infected USB flash drive from a compromised equipment vendor. Secondly, it spread locally through the SCADA network in three ways: using the normal LAN, via removable drives, and by infecting files used by Siemens PLCs. The objective of this phase was to look for computers possessing the Siemens WinCC SCADA software, typically used to program PLCs, and to establish a foothold on those machines. The third and final stage probed for PLCs connected to the WinCC system: once found, malicious code was injected to stealthily control specific centrifuges, making them operate at unsafe speeds and resulting in a higher breakdown rate [24].

Dragonfly. The Energetic Bear/Dragonfly campaign of 2011 focused on industrial espionage and intellectual property theft rather than taking control of the industrial process. It specifically targeted industrial gateways and routers used in aviation, energy generation and distribution, pharmaceutical, food and beverage industries [21].

The infection happened in three phases [26], [4]: initially, the attackers delivered malware through spear-phishing emails; then, they performed a watering hole attack by redirecting traffic from legitimate websites; and finally, they infected third-party applications that ICS device vendors made available online, thus compromising the supply chain. The malware then communicated to a command and control (C&C) server via HTTP, downloaded additional modules establishing persistence, and scanned the local drives collecting information about the network layout, as well as ICS and VPN configuration files, and authentication credentials. It did not spread over the local network. Its final stage was to use an industrial protocol scanner to search the local network for any OPC services (see Fig. 1), or for devices and applications that were listening on TCP ports of common SCADA protocols. A compromised OPC could have granted an attacker full control over the SCADA system, but the attackers made no attempt to control the ICS devices: instead, the gathered data about the SCADA network layout was sent back to the C&C server.

BlackEnergy. In late 2015, three Ukrainian power distribution utilities suffered a coordinated attack that caused a blackout for several hours [32].

The attack was conducted in two main stages, separated by months [12]: first, the attackers used phishing emails to penetrate the utilities' IT networks and plant the BlackEnergy 3 malware. The malware connected to its C&C server, moved horizontally and harvested credentials to gain VPN tunnelling access to the SCADA network; once there, it completed the initial reconnaissance by discovering the serial-to-ethernet field devices used by the remote stations to decode commands from the command center. Six months later, the attackers used the malware to take control of the SCADA workstations and HMI, locking out operators and manually issuing commands to open the remote stations' breakers, thus causing the blackout. At the same time, they deployed malicious custom firmware on the gateway devices, disabling them and preventing recovery.

VI. Where encryption fails

With basic definitions and a description of key attacks in place, we can now evaluate our first thesis: *encryption often does not yield extra SCADA security.* To this end we consider the impact of encryption on the attacks described above.

Stuxnet. Recall that Stuxnet comprises three stages. The first stage, i.e. the initial infection through a compromised USB drive, did not involve network communication. In the second and third stages, Stuxnet first spread on the LAN and then infected WinCC database servers; the infected WinCC systems then uploaded control code to the PLCs, as they were authorized to. However, this code had malicious content. In both stages, all communication were between valid parties that trusted each other. The endpoints' vulnerabilities exploited in the second stage to spread Stuxnet, and the malicious content transmitted to PLCs during the third stage, did not affect the proper establishing of the connections. As such, encryption wouldn't have impeded the attack at all.

Dragonfly. The Dragonfly campaign used standard business level malware techniques, focused on the target's corporate network [21]. Once there, the malware gathered locally stored authentication credentials that enabled authorized access to other remote industrial systems. In around 5% of the infections, the malware included a module to capture credentials sent over unencrypted HTTP traffic from a browser[4], [3]. Also, the attackers tried to discover and probe OPC services on LAN hosts, by using the valid interfaces that were already present on the infected machines. The situation was the same as with Stuxnet, in that the attackers exploited vulnerabilities on the end points, while all the communications on the network was between valid parties. Only in some rare cases, encryption would have hindered a small portion of the information gathering performed by the malware.

BlackEnergy. The attackers infiltrated a business workstation through email, spread their malware on the LAN, and then harvested credentials to gain legitimate and authorized access to the SCADA network, bypassing the security at the gateways of the remote stations. Using existing remote administration tools, the attackers used "native connections and commands" [12] to discover the ICS devices on the remote stations' local networks; to upload the custom malicious firmware to the gateways; and to control the breakers through a panel. All these malicious actions compromised endpoints rather than connections, and therefore would not have been impacted by encrypting SCADA traffic.

As stated before, encrypting a communication channel protects the confidentiality of a message during its transmission. This is relevant in the case where potential attackers reside along the transmission path of the message, either intercepting it as a Man-in-the-middle or just passively listening to it. On the other hand, if the attackers compromise

Brief Description	Encr.	Net Mon.	Year	Industry
Stuxnet Malware Targets Uranium Enrichment Facility [1], [14]	X	O f,c [24]	2010	Power/utility
Russian-Based Dragonfly Group Attacks Energy Industry [1], [4]	X	O f,c [22]	2014	Power/Utility
Cyber-Attack Against Ukrainian Critical Infrastructure [1], [12]	X	O f,c [32]	2016	Power/Utility
Malware on manufacturing OT network [31]	x	O f,c [31]	2017	Manufacturing
Hacktivist control PLCs of "Kemuri Water Company" [30]	x	o c	2016	Water treatment
Public utility compromised after brute-force hack attack [1]	x	o f,c	2014	Power/utility
U. S. Power Plant Infected With Malware from USB [1]	x	?	2012	Power/utility
U. S. Electric Utility Mariposa Virus Infection [1], [16]	x	O f,c [16]	2012	Power/utility
Disk-wiping Shamoon virus knocks out computers at Qatari gas firm RasGas [1]	x	?	2012	Petroleum
Gas Company Virus Infection from USB [1]	x	?	2012	Petroleum
Auto Manufacturer Suffers Data Breach from Virus [1]	?	?	2012	Automotive
Process Control Network Infected with a Virus from Laptop [1]	x	?	2012	Petroleum
Industrial Control System Hacked Using Backdoor Posted Online [1], [15]	x	o f,c	2012	Other
South Houston Water Treatment Plant Hack [1]	x	?	2011	Water/Waste
Steel plant infected with Conficker Worm [1]	x	o f,c	2011	Metals
Brute-Force Attack on Texas Electricity Provider [1]	x	o f	2010	Power/utility

TABLE I
ANALYSIS OF RECENT SCADA INCIDENTS

a communication endpoint, as it happened in our examples, it's easy to obtain the keys and configuration files to establish valid connections to other devices in the SCADA network, and pivot the attack to those.

The second column of Table I summarizes the evaluation of the different attacks. For the three attacks studied in detail, encryption did not help (indicated by 'X' in the table). The same conclusion can be reached for the others, based on a general description of the attack (indicated by 'x'). In one case (indicated by '?') we did not have enough information on the attack to evaluate whether encryption would have helped. The table clearly validates our first thesis; encryption is not able to stop most of these attacks.

VII. Threats of encryption to security

In this section we evaluate our second thesis: *Encryption can have negative consequences for security.* Encryption decreases visibility of data, not only for potential attackers, but also for security tools trying to evaluate this data such as network monitoring solutions. With respect to monitoring we distinguish two main categories; flow-based solutions e.g. [28] that only consider the amounts of communication and the end-points involved, and content-based solutions e.g. [13], [35] that also consider the actual content of the communications. Flow-based solutions may still work if the communication is encrypted, but this depends on the exact approach and the method of encryption. IPSec tunnel mode, for example, would prevent (some forms of) flow-based analysis on the link it is applied on. Clearly, content-based solutions would be prevented from fully analysing data that is encrypted with keys the monitoring system does not have.

In the third column of Table I we indicate whether the attacks could have been detected by network monitoring, distinguishing between cases (marked 'O') where detection is certainly possible, as reported by the indicated publications, and cases (marked 'o') where we believe detection should be possible based on a high level evaluation of the attack. All three attacks discussed in Section V could have been detected by an appropriate network monitoring solution. We further indicate whether flow-based (f) and content-based (c) monitoring is involved. Several attacks (marked f,c) can be detected by flow-based monitoring but require content-based approaches to identify what type of attack is happening.

We have several cases where we did not find any claims that the attack is detectable with a given approach, and the attacks' descriptions are not sufficient to determine whether known approaches would work. As such, there are several cases that are indicated as unknown (?). Still, several attacks require content-based approaches to identify or even to detect them at all. This already validates our second thesis; in many cases encryption hinders other security solutions and thus may actually decrease the security of the system.

VIII. Threats of encryption to system operations

In this section we evaluate our third thesis: *Encryption increases troubleshooting and recovery costs.* To this end we consider several causes that can motivate troubleshooting.

Network congestion. Upon slow operator terminal updates one would check the LAN for overload [9, Sec. 8.2].

Quoting from [31]: "over the past few months, the network seemed 'sluggish', which the automation engineers and SMEs attributed to older, legacy equipment. [...] With the cooperation of [company], we set up a Switched Port Analyzer (SPAN) port and deployed a passive network analyzer to collect and analyze the traffic." If the traffic was encrypted, this common troubleshooting task would have been hindered.

A possible cause for congestion is a device flooding the network, e.g. due to misconfiguration or a virus attack. An example of the latter was the Conficker worm infecting a steel plant in 2011 [1]: "The virus flooded the network with unwanted packets and caused an instability in the communications between PLCs and supervisory stations and freezing most of the supervisory systems." While the presence of the flaw is clear, a full diagnosis requires looking at the content of the communication and possibly listening from different locations, to identify the source of the anomalous traffic.

Non-healthy devices. Upon missing updates, alarms or unexpected behaviour one would evaluate the health of related components. After basic (hardware) checks, [9, Sec. 6.10] recommends checking an individual component's health by

using a protocol analyser to look for errors or inconsistencies in its traffic. Components failing health tests should be readily replaced: "An effective SCADA system should include the proper complement of spare components that the operator can swap out easily for troubleshooting purposes." [2] The lower visibility of data induced by encryption can negatively affect these health checks, and key management issues can impact prompt replacement of components.

Third-party network access. A common SCADA practice is to hire an external party to evaluate the system, either as part of a health check or risk assessment [31], or for emergency troubleshooting. As part of this, the external party would plug a (possibly unauthenticated) external device (laptop) at different points of the communication network, and evaluate the systems and communication visible there. As even authenticated devices do not normally get the decryption keys for sessions between other devices, encryption might hinder this practice by limiting what communications are visible to the external device.

The examples above show that encryption increases troubleshooting complexity by making analysing problems and replacing components more involved. The exact impact may differ per scenario; a more formal general statement would require going into SCADA troubleshooting and recovery common practices in detail. Still, we believe the issues observed above are representative and confirm our thesis that encryption increase troubleshooting and recovery costs.

IX. Conclusions

This paper is meant as a critical analysis of the pro's and con's of network encryption for ICS. We observed three general principles: First, in the majority of cases, the introduction of encryption does not yield extra security. Second, encryption can actually have negative consequences for security by hindering other security mechanisms such as NIDS. Third, encryption can raise the costs of troubleshooting and recovery considerably. Of course, before drawing conclusions one has to consider the criticality of the target ICS, as well as its specific requirements: for example, systems dealing with user data such as advanced metering infrastructures (AMI) will need stronger confidentiality. Currently, though, in typical ICS scenarios one needs to achieve authentication and integrity of the communication (whose implementation is easier and has less impact on the general system), rather than the confidentiality offered by encryption. We cannot predict any new attacks or future changes to the threat landscape that might change this priority.

We do not advocate for completely discarding encryption for ICS network traffic, but assert that blanket use of encryption on SCADA networks can prove both costly and detrimental to security. Instead, careful consideration of what encryption is actually good for, and at what cost, is needed both for standardization efforts, and SCADA system deployment.

[2] www.tpomag.com/online_exclusives/2013/07/scada_troubleshooting_tips_help_systems_run_smoothly

References

[1] RISI Online Incident Database. http://www.risidata.com/Database.

[2] APT1: Exposing One of China's Cyber Espionage Unit. Technical report, Mandiant, 2013.

[3] Cyberespionage attacks against energy suppliers, version 1.21. Technical report, Symantec, 2014.

[4] Energetic Bear - Crouching Yeti. Technical report, Kaspersky, 2014.

[5] Annual Threat Report. Technical report, Dell, 2015.

[6] *IEC 62351: OPC Unified Architecture*. International Electrotechnical Commission, 2015.

[7] Year in Review. Technical report, NCCIC/ICS-CERT, 2015.

[8] *IEC 62351 (2016-09): Power systems management and associated information exchange - Data and communications security*. International Electrotechnical Commission, 2016.

[9] David Bailey and Edwin Wright. *Practical SCADA for industry*. Newnes, 2003.

[10] Stewart Baker, Shaun Waterman, and George Ivanov. In The Crossfire. Technical report, McAfee, 2010.

[11] Manuel Cheminod, Luca Durante, and Adriano Valenzano. Review of security issues in industrial networks. *IEEE Transactions on Industrial Informatics*, 9(1):277–293, 2013.

[12] Tim Conway, Robert M. Lee, and Michael J. Assante. Analysis of the cyber attack on the Ukrainian power grid. Defense use case. Technical report, SANS ICS, 2016.

[13] E. Costante, J.I. den Hartog, M Petković, S. Etalle, and M. Pechenizkiy. Hunting the unknown - white-box database leakage detection. In *DBSEC, LNCS 8566*, pages 243–259, 2014.

[14] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 5(6), 2011.

[15] FBI. Vulnerabilities in Tridium Niagara Framework Result in Unauthorized Access to a New Jersey Company's ICS, 2012.

[16] ICS-CERT. Advisory ICSA-10-090-01: Mariposa Botnet, 2010.

[17] S. Kent. IP Authentication Header. RFC 4302, 2005.

[18] S. Kent. IP Encapsulating Security Payload (ESP). RFC 4303, 2005.

[19] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301, 2005.

[20] HyungJun Kim. Security and vulnerability of SCADA systems over IP-based wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2012.

[21] Joel Langill. Defending Against the Dragonfly Cyber Security Attacks. Technical report, Belden, 2014.

[22] Joel Langill, Emmanuele Zambon, and Daniel Trivellato. Cyberespionage campaign hits energy companies. Technical report, Security Matters, 2014.

[23] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.

[24] Ralph Langner. To Kill a Centrifuge. Technical report, Langner Group, 2013.

[25] David McMillen. Security attacks on industrial control systems. Technical report, IBM, 2017.

[26] Nell Nelson. The Impact of Dragonfly Malware on Industrial Control Systems. Technical report, SANS ICS, 2016.

[27] Adam Slagell. Thinking critically about computer security trade-offs. *Skeptical Inquirer*, 2016.

[28] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller. An overview of ip flow-based intrusion detection. *IEEE Communications Surveys and Tutorials*, 12(3):343–356, 2010.

[29] Keith Stouffer, Suzanne Lightman, Victoria Pillitteri, Marshall Abrams, and Adam Hahn. *Guide to industrial control systems (ICS) security*, volume 800. NIST, 2014.

[30] Verizon RISK Team. Data breach digest, 2016.

[31] Verizon RISK Team. Data breach digest, 2017.

[32] Daniel Trivellato and Dennis Murphy. Lights out! Who's next? Technical report, Security Matters, 2016.

[33] Yongge Wang. sSCADA: securing SCADA infrastructure communications. *Int. J. Communication Networks and Distributed Systems*, 6(1):59, 2011.

[34] Wonderware Invensys Systems. *Securing Industrial Control Systems*, 1.4 edition, 2007.

[35] Ömer Yüksel, Jerry den Hartog, and Sandro Etalle. Towards useful anomaly detection for back office networks. In *ICISS, LNCS 10063*, pages 509–520. Springer International Publishing, 2016.

[36] Bonnie Zhu, Anthony Joseph, and Shankar Sastry. A taxonomy of cyber attacks on SCADA systems. In *iThings/CPSCom*, pages 380–388. IEEE, 2011.