A Study on Vulnerabilities and Threats to SCADA Devices

Dawn Silverman
Department of Computer Science
Norfolk State University
Norfolk, VA, USA
d.m.silverman@spartans.nsu.edu

Yen-Hung (Frank) Hu
Department of Computer Science
Norfolk State University
Norfolk, VA, USA
yhu@nsu.edu

Mary Ann Hoppa
Department of Computer Science
Norfolk State University
Norfolk, VA, USA
mahoppa@nsu.edu

***Abstract*—SCADA devices have increasingly become targets of malicious actors, alerting industries, governments and even private citizens to the need for more effective security measures, particularly for critical infrastructure and industrial control systems. To address concerns on this issue, a thorough survey and investigation was conducted on cyber-attacks targeting SCADA systems to propose solutions and recommendations for mitigating such attacks. This research first studied some historical perspectives on SCADA and associated risks, including examples of typical attacks. After summarizing known SCADA vulnerabilities and some attempts to harden these systems, a deeper-dive was taken on a breach of the Schneider Triconex Tricon 3008 safety system as an instructive use case. Some general recommendations were made for methodically securing SCADA networks. The long-term objective of this research is to better secure the future of SCADA and, by implication, the critical infrastructures that depend on this technology, through more focused cybersecurity vulnerability assessment and mitigation.**

Keywords—Industrial Control System, SCADA

I. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) technology has evolved since its inception in the 1950's through four generations [1]. Each generation has brought new capabilities and a more scalable network as the performance of SCADA devices have increased through the years. With each generation has come different vulnerabilities along with new challenges and opportunities to securing SCADA-enabled systems from malicious attacks.

Critical infrastructure and industrial control systems (ICS) throughout the United States (U.S.) and abroad are dependent on SCADA devices to automate and monitor processes. They enable the delivery of essential services including electric power and purified water to residential and commercial properties, where a sustained interruption could create a crisis with dire consequences. As one example, chronically ill residents reliant on durable home medical gear such as oxygen concentrators and kidney machines could suffer loss of life.

Major functions of SCADA include remotely monitoring many processes, collecting real-time critical data, and performing data analysis. SCADA systems usually consist of three main components – hardware, software and communications interfaces; these will be discussed in detail later. Typical SCADA system configurations involve a central host computer, a number of remote terminal units (RTUs), operator terminals, human-machine interface (HMI) software, and devices such as sensors, valves pumps, and motors [2]. The HMI provides a means of controlling all the devices attached to it. Thus within SCADA systems, the HMI represents the most valuable target for attackers. By successfully gaining access to this software, an attacker virtually owns that SCADA network [3].

Legacy SCADA systems, designed and implemented decades ago, were deployed with a "security through obscurity" mentality. That is, physical isolation, proprietary protocols and technical uniqueness were assumed to deliver a secure solution inaccessible to unintended actors. While originally this may have provided sufficient security, with the development and expansion of the internet, SCADA devices are increasingly employed in distributed, open architectures and accessible via corporate networks. Their proprietary protocols and technologies have not been modernized, or have given way to open standards. This has left SCADA vulnerable to a wide variety of cyber-attacks, ranging from simple password hacks to more sophisticated internet espionage and Advanced Persistent Threats (APTs) [4].

SCADA devices have become more vulnerable through the years as their accessibility via the internet and corporate networks has grown. Another exacerbating factor is continued use of legacy software and firmware that are no longer being patched by vendors, such as obsolete versions of Microsoft Windows. Additionally, studies have shown that half of all industries reliant on SCADA technology do not even have minimal countermeasures in place such as anti-virus protection [5].

Modern tools used to perform security audits and penetration tests are now being used on older SCADA networks. Without careful configuration, these tools can cause significant damage to SCADA devices connected to a corporate infrastructure, rather than helping to protect and audit them. In addition, the sharing of device specifications and manuals online, coupled with the ease of scanning SCADA systems, can help bad actors visualize the infrastructure, which in turn may enable their malicious activities [6].

Through the years SCADA device functionality and the need for real-time business information from any location

979-8-6425-4896-7/20/$26.00 ©2020 CISSE

have created another vulnerability issue. While previous versions of SCADA systems were standalone, now these systems also are connected to other systems not directly related to process control and monitoring. In an effort to reduce cost and improve performance, both ICS vendors and owners have been transitioning from more proprietary solutions to the less expensive technologies prevalent today, such as Ethernet, TCP/IP, and Microsoft Windows [7] [8].

As additional ICS components become more interconnected with the outside world, the likelihood and potential impact of cyber-attacks will heighten [7] [8]. So much information can be found online; even the flaws in SCADA specific technologies have become readily available to the public. This poses an even greater risk. As a basis for identifying opportunities to close the gap on SCADA vulnerabilities, this research explored areas of SCADA systems that should be subjected to more focused vulnerability analysis to guide the development of methodologies that can be useful to better secure them.

The remainder of this paper is organized as follows: Section 2 summarizes related work and recent government efforts to provide perspective on the scope and importance of the SCADA problem. Section 3 discusses known examples of SCADA vulnerabilities and attacks. Section 4 discusses general SCADA vulnerabilities in hardware, software, communications and standard operating procedures (SOPs). Section 5 introduces a real SCADA use case: vulnerabilities in the Schneider Triconex control system. Section 6 proposes potential resolutions to the four key SCADA vulnerability areas. Section 7 concludes the paper with some reflections on findings and suggestions for future work to build upon them.

II. Related Work

While the potential for threats against SCADA assets, particularly utilities, has been recognized for decades, since the September 11, 2001 terrorist attacks the security of ICS and critical infrastructures has come under intense scrutiny. Moreno gives a quick overview of SCADA technology for those new to the topic [10], while Igure et al. have explored crucial research issues involved in strengthening cybersecurity in SCADA networks [9]. Ten *et al.* considered problems through the lens of electric power systems. They proposed a framework for SCADA vulnerability assessment at three levels: systems, scenarios and access points [11]. Attack trees are another popular assessment methodology that has been applied to both SCADA specifications and deployments [12]. Coffey et al. speculated that due to the bespoke characteristics and purposes of SCADA equipment, inspecting them with certain tools used in vulnerability assessments – such as asset discovery, service detection, and network scanners – may negatively impact how they operate, and even open up new vulnerabilities [6]. This suggests that modeling and virtualization may be recommendable for building SCADA vulnerability assessment platforms similar to [13] and [14].

In 2001 President Bush issued Executive Order 13231, which coordinated all Federal activities related to the protection of information systems and networks supporting critical infrastructure [15]. In fulfillment of this initiative, the Secretary of Energy's Office of Independent Oversight and Performance Assurance has conducted a number of assessments of organizations with SCADA networks to develop an in-depth understanding of them. With that understanding, they were better able to recommend necessary steps to secure SCADA networks.

As the potential for malicious attacks on SCADA systems within the U.S. and other countries came increasingly into focus for experts, industries, politicians and average citizens, President Barack Obama decided that cybersecurity had to be at the forefront of the agenda and governed at the Federal level. The U.S. Government therefore enacted a series of Executive Orders – 13687, 13691, and 13694 – that comprise first steps toward securing against cyber threats [16]. These orders directly affect the security of SCADA devices, since some areas of SCADA networks may be accessible through the internet. Moreover, ongoing investigations into the 2016 U.S. Presidential election hack [17], and the order to secure critical infrastructure sectors [18] [5], also are relevant to SCADA systems.

III. SCADA Vulnerability Trends

A recent study [5] found the following vulnerabilities are prevalent among many SCADA systems:

- Connectivity to the public internet
- Unpatched legacy software (e.g., Windows XP and 2000)
- Weak authentication
- Lack of antivirus software
- Rogue (unauthorized, unrecognized) devices
- Presence of undetected malwares and APTs
- Remote management protocols
- Wireless access points
- High proportion of vulnerable devices

Trend Micro's Zero Day Initiative (ZDI) [3] examined the state of SCADA HMI devices in particular, and reported the most exploited vulnerabilities are:

- Memory corruption (20%)
- Credential mismanagement (19%)
- Lack of authentication/authorization and insecure defaults (23%)
- Code injections (9%)
- Other means (29%)

SCADA StrangeLove [19] is a group of security researchers focused on preventing industrial disasters. Since 2012, they have maintained a web presence and reported over 150 zero-day vulnerabilities in ICS and programmable logic controllers (PLCs), with five percent of these being dangerous remote code execution attack vectors.

As the Internet of Things (IoT) becomes pervasive in everyday life, so too are more ICS and infrastructures connecting to the internet. Thus, no sector can persist in thinking that SCADA risk is someone else's problem. The current generational evolution is starting to expand SCADA into the cloud, where the capacity to share massive amounts of data via wireless technology brings new possibilities for cost reduction and reliability to industries, but also offers more motivation and attack vectors to cyber criminals [20].

A recent investigation of SCADA devices with embedded operating systems (OSs) discovered over 10,000 of them are accessible via the internet and lack strong authentication controls [21]. This provides an entrée for cybercriminals to analyze ports and to use hardware hacking techniques, such as firmware dumping and reverse engineering, to determine how each device works and how it can be attacked. A number of published attack examples [16] [22] [23] [24] has served to alert industries, governments and even private citizens to the abundance and range of SCADA vulnerabilities, which all too often have set a low bar for malicious actors who are intent on disrupting critical infrastructure.

IV. Key SCADA Vulnerabilities

In light of the above trends regarding SCADA vulnerabilities, this section takes a more methodical look at them from four perspectives: hardware, software, communications and standard operating procedures (SOPs).

A. Hardware Vulnerabilities

Hardware vulnerabilities occur in such components as RTUs, HMI, PLCs, and smart devices that report back to the main terminal in a SCADA system.

The typical SCADA environment uses an HMI by which all smart devices are monitored. In today's SCADA networks, these HMIs are highly advanced and can be customized to monitor a system's current state. Information provided to operators may include the state of control systems and specific sensors. The HMI also provides a means to facilitate any corrective measures that may need to be undertaken. HMIs are a primary target within SCADA systems, which suggests they should be air-gapped or isolated on a trusted network due to the vulnerabilities they create [3].

RTUs within a SCADA system usually are centrally controlled by a master system. Typically, RTUs consist of devices such as relays, actuators, circuit power breakers, voltage regulators, and a multitude of sensors. SCADA environments are interconnected to the master station through a variety of channels and means, such as radio links, leased lines, and fiber optics, all of which contribute their own vulnerabilities [25]. PLCs contribute additional vulnerabilities to the SCADA environment [26]. They have limited capabilities for implementing advanced control algorithms. In a critical infrastructure vulnerability test, 1,207 out of 1,843 Allen-Bradley Micro Logix 1400 PLC devices (65 percent) were found to have critical vulnerabilities. This puts them at risk for buffer overflows and Man-in-the-Middle (MitM) attacks that can harm both devices and the infrastructures they control [20].

B. Software Vulnerabilities

One of the most important elements of a cybersecurity attack is the software. Each year the number of known vulnerabilities in software grows. This results in more potential for malicious attacks from hackers. Software attack statistics are maintained by the Computer Emergency Response Team/Coordination Center (CERT/CC) and the US-CERT. The statistics from these organizations show that the number of known OS vulnerabilities and security holes in software technology has significantly increased over the decades. And these statistics are not even complete because many organizations are reluctant to publicly disclose their statistical data about intrusion attempts [11].

The use of an embedded OS requires additional expense and effort because it is tougher to interact with and maintain. This is one of the main reasons more common OSs tend to be used in SCADA control center systems, such as Microsoft Windows, UNIX, and Solaris [25]. Consequently, SCADA environments are subject to the same broad variety of vulnerabilities found in these OSs. Another category of software vulnerability inherited by SCADA environments derives from the multitasking and real-time databases and servers they use for data acquisition and tracking the many parameters being monitored [25].

According to [25], some key SCADA software security issues include the following:

- Viruses, malware, and Trojan horses: Malicious content is introduced by a variety of means including opening infected attachments; clicking on links from unknown or spoofed emails; and downloading fallacious software updates or patches. Failure to install legitimate software updates when they become available exposes systems to attack risk.
- Logical errors: Code flaws may be created during system development and can cause unintended or undesired side effects. Logical errors may not be immediately known to the user until an unexpected or incorrect result is produced. "Zero-day" vulnerabilities can result from flaws that create security holes and are not yet known to the developers.
- Convenient features for users: Infections can result from features used by most users. One example is messaging based on Simple Mail Transfer Protocol (SMTP) that was developed for users' convenience. This standard e-mail system suffers numerous vulnerabilities that may be activated by users' actions, such as email spoofing, eavesdropping, and malicious content downloads.
- Authentication permissions: Typically, system administrators are provided an interface to manage

user credentials and to make other system-wide changes. Devices and applications may ship with well-known default accounts and credentials in place. Administrators sometimes fail to change the defaults so credentials are easy to remember, which leaves the door open to hackers. Individual users' access permissions within SCADA systems may not be assigned in alignment with the principle of least privilege either, which can allow them unnecessary access to sensitive areas where they can intentionally or inadvertently do harm.

- Administrator access: The creation of administrator accounts with promiscuous or unrestricted accesses and permissions (so-called "super users") paves the way for cyber-attacks involving privilege escalation, and increases opportunities for insider threat activities.

C. Communications Vulnerabilities

A typical SCADA communications system consists of a master station and many other distributed RTUs. These RTUs are interconnected to master stations through a variety of communications channels and protocols [25]. One of the greatest challenges is that the channel limits the speed of data acquisition control that can be performed. Random noise on the channel is another challenge that has plagued SCADA communications. The interconnection of microprocessors used in SCADA has been an increasing trend, and this interconnection creates even higher security risks for SCADA systems [25].

Whereas legacy SCADA devices were implemented to be isolated and stand-alone, the new generation of SCADA devices can be accessed from anywhere in the world. Some legacy devices are still in use today and have not been upgraded, patched, or otherwise ruggedized for the complexities and threats in today's cyberspace. Communications between these devices leaves them open for attacks. Successful MitM exploits can reroute communications to a malicious actor who wants to "own" the network. Denial-of-service (DoS) attacks can severely diminish how SCADA hardware operates and the integrity and timeliness of reports back to the main unit.

D. Procedural Vulnerabilities

Policies and procedures – often referred to collectively as standard operating procedures (SOPs) – are at the root of every successful security program. SOPs help ensure that security mechanisms, decisions and actions are both consistent and current to protect against malicious attacks. According to the National Institute of Standards and Technology (NIST), SOPs should focus on systems holistically rather than just individual devices, and should include PLCs, Distributed Control Systems (DSCs), SCADA, and instrument-based systems that use a monitoring device such as an HMI [27].

If SOPs are not regularly re-evaluated, they may not include the most up-to-date information for securing a SCADA environment. They also may fail to identify and address new or deprecated devices, applications and computer systems associated with the SCADA architecture. Legacy SOPs may omit security best practices considered to be "basic" by today's standards, such as limiting access paths and creating a physical gap between the SCADA systems and the business network. Other oversights may include lax identity management and administrator accounts with well-known default credentials that are easy to hack. An absence of encryption protocols for master and slave device communications, and a lack of advanced authentication techniques such as multi-factor and biometrics, are more red flags that the latest security technologies have not been integrated into the SOPs [28]. Overlooking such details provides avenues for attackers to penetrate into the SCADA environment.

V. SCADA Vulnerability Use Case

To identify additional areas of SCADA that should be scrutinized more closely, it can be useful to examine the details of breaches that have been shared in the public domain. One such use case is a petrochemical plant in the Middle East whose safety system was attacked in late 2017, resulting in a multi-hour plant shutdown.

This section will analyze information from various published case studies for hardware, software, communications and SOP vulnerabilities related to a hack on a device called the Triconex 3008 manufactured by Schneider Electric. The malware used in this attack – called Trisis, Triton, or HatMan – targeted a safety shutdown system by replacing logic in a final control safety element in a SCADA environment. It is the first publicly known example of malware that specifically targeted an ICS [30].

Walking through this incident shows that simply following the manufacturer's instructions could have prevented the breach from occurring.

A. Schneider Triconex 3008 Safety Controller

The safety system attacked in this case consisted of the Triconex 3008 running the TriStation 1131 software. The Triconex 3008 is a safety control system main processor used for control programs, sequence-of-events data, input/output (I/O) data, diagnostics and communications buffers. In the event of an external power failure, the integrity of the user-written program and the retentive variables is protected for a minimum of six months. The main processor modules receive power from dual power modules and power rails in the main chassis. A failure in one power module or power rail will not affect the performance of the system [31].

B. Vulnerabilities in Triconex Hardware

The legacy Tricon controller involved in this incident has a physical key switch that is turned to put the system into different modes: the "program mode" allows logic changes; whereas the "run mode" prohibits them and is the intended setting for when the system is operational. The system instead had been left in the "program mode" during operations, which exposed it to scanning and commands issued by malware. Schneider Electric rightly concluded that the system had been working properly; had it been correctly set to “run mode” instead of “program mode,” the malware could not have succeeded.

An additional identified vulnerability is that all Tricon controllers are shipped with identical keys, and there is no procedure in place for a customer to order a different key for their systems [32]. This makes all Tricon controllers vulnerable to compromise due to key loss, key theft, disgruntled insiders or former employees who may have copied or stolen a key.

Researchers at Dragos have laid out alternative architectures and explanations for safeguarding these safety security controllers [29].

C. *Vulnerabilities in Triconex Software*

The Triton malware used in this attack gained remote access to a Triconex engineering workstation running Microsoft Windows and DCS. It reprogramed the safety controller using the TriStation software used to run the system.

The attacker deployed a Py2EXE application, which was disguised as a benign Triconex log reviewing application named Trilog.exe, containing the Triton framework on the engineering workstation together with two binary payload files named inject.bin and imain.bin [30].

The TriStation software is proprietary and undocumented. It is speculated the hackers reverse-engineered its protocols by mining the documented Triconex System Access Application (TSAA) protocol [30]. The TriStation protocol is typically set up as User Datagram Protocol (UDP)-based serial over Ethernet. UDP is an alternative communications protocol to Transmission Control Protocol (TCP). It is used primarily for establishing low-latency and loss-tolerating connections so it is a standard in the ICS world. The request packets contain a two-byte function code (FC), which is then followed by a counter identifier, length field and request data together with checksums. The Triton attack framework leveraged a sequence of these function codes and expected response codes [30].

The effects of the Triton malware can be thought of as a four-stage shellcode. A shellcode is a list of instructions that can be executed once the code is injected into a running application. The first stage of this malware is an argument-setting piece of shellcode. The argument-setter is a value that is passed between programs, subroutines or functions. They are independent items, or variables that contain data or codes. The second is formed by inject.bin, not currently available. This inject.bin functions as an implant installer. The third stage is formed by imain.bin. This functions as a backdoor implant that is capable of receiving and executing the fourth stage. The fourth and final stage of this malware would have been formed by an actual OT payload performing the disruptive operations. Apparently no such payload was recovered during the incident since the attacker was discovered while preparing the implant of this malware [30].

The TriStation Developer’s Guide mentions it is possible to restrict access to a Tricon controller from a TriStation PC. Projects set up using the TriStation software automatically create an administrative account with the highest level of privileges; by default, the user name is “MANAGER” and the password is “PASSWORD” [33]. Many times, default user names and passwords are never changed by end users so they are easy to remember and manage. To make matters worse, by default no password is needed to connect to the controllers themselves (although this setting can be changed).

Since the TriStation Developer’s Guide is posted online and available to anyone, unchanged default credentials pose a major vulnerability. The TriStation protocol itself is unencrypted; therefore, any MitM attacker observing network traffic between the controller and the TriStation workstation can circumvent authentication protections anyway [30].

Triconex is working on a solution to this malware, and the U.S. Department of Homeland Security is heading an investigation into the matter.

D. *Vulnerabilities in Triconex Communication*

The Triconex industrial safety controller has many different communications modules to facilitate serial and network communications across a variety of protocols. One example is the Tricon Communication Module (TCM) which allows communications between a controller using the TriStation 1131 software. This can be configured to use Modbus master/slave for devices and external hosts over Ethernet networks [30].

In one case study [34], the Modbus protocol was used to analyze a real-time vulnerability because:

- Modbus is still widely used in SCADA systems.
- Modbus/TCP is simple and easy to implement.
- Modbus protocol libraries are freely available for utilities to implement smart grid and SCADA applications.

Wireshark was used to analyze the network traffic. Two well-known attacks were performed in a test bed: DoS and MitM. Both attacks severely impacted system operation and stability. This study also revealed that the Modbus protocol has lax security, with no access control lists and no form of trust domain [34]. Thus, it is not a wise choice for SCADA environments.

E. *Vulnerabilities in Triconex Standard Operating Procedures*

A failure to follow proper operational procedures – in this case, the selection of an inappropriate key-switch setting – left the Triconex open to attack by the Triton malware. The capability to configure the TCM to use Modbus with its lax security protocols was not specifically cited as a root cause for the Triton attack, but points to a general procedural gap between what is technically possible to do, and what is recommendable to do when security is paramount.

Unrestricted posting of high-value documentation artifacts for SCADA devices – such as Developers' Guides and other specification details – would have been less likely had security-facing best practices been in effect for both manufacturers and users.

Some decisions and actions are side-effects of dated or insufficient SOPs and have the potential for grave impacts on system and network security. Examples in this use case include the unencrypted Triton protocol, a legacy design choice that is insufficiently secure for use in modern environments. There was no documented mechanism in place to provide for unique device keys and to exert positive physical control over their whereabouts, putting them at risk for loss, theft and therefore unauthorized use.

VI. Recommendations for Key SCADA Vulnerabilities

Due to their importance to ICS, critical infrastructure and quality of life, SCADA networks must be safeguarded. While there is no such thing as a perfectly secure system or network, by coalescing reports in the literature and the details of a specific use case, this research has uncovered a number of precautionary measures that can help mitigate some vulnerabilities and thereby impede malicious actions against SCADA. These recommendations are grouped into four areas common to all SCADA systems and where specific actions can be undertaken to harden them against cyber-attacks: hardware, software, communications, and SOPs.

A. Hardware Recommendations

In today's SCADA environments there are many legacy devices in use that can no longer be upgraded or patched for vulnerabilities. Such devices should be removed or replaced immediately. HMI systems are the most vulnerable and prized by attackers and therefore would benefit from being air-gapped and isolated from the rest of the network. SCADA hardware devices always should have physical controls limiting who can access them, and if they are in remote locations they should be monitored and locked.

B. Software Recommendations

The most important recommendation for all software used in SCADA networks is keeping it up-to-date and patched. New version releases and patches generally improve functionality and security features; installing them helps protect the network from the latest known threats. It also is important to confirm that all upgrades, updates and patches are from authenticated providers, not spoofed websites trying to deliver surreptitious malware payloads to penetrate the SCADA network. As mentioned earlier, outdated and proprietary SCADA devices running code that is no longer being patched can pose a significant vulnerability. Such software and devices should be prioritized for replacement or deprecation.

Using antivirus software on any network reduces the possibility of malicious content infiltrating and causing harm to devices. Because SCADA devices typically are used in critical infrastructure, they especially should be configured using highly effective antivirus software. Antivirus software should be updated often to protect against the latest malware trying to penetrate SCADA networks.

The use of an embedded OS in a SCADA network decreases the likelihood of attacks because it is tougher to interact with such systems. After software installation, permissions should be set to the highest practicable level for added security. The principle of least privileges should be enacted for all accounts on all SCADA networks, since this may stop a malicious attack from privilege escalation. Implementing strong authentication controls, including two-factor authentication or better, will add an extra layer of protection.

C. Communications Recommendations

Due to the sensitivity of the operations they support, there may be multiple different security trust levels within a SCADA network. A baseline of acceptable, "normal" use and traffic should be established and monitored. As a defense mechanism, firewalls can be used for this purpose by filtering the bi-directional packet flows within and between networks to help manage incoming and outgoing traffic. Firewall filter criteria should be established in consonance with the baselines, such as expected protocol types, port services or port service ranges, and Internet Protocol (IP) addresses or ranges [11]. A firewall can be implemented for a SCADA network either by connecting external hardware, or by integrating software into the SCADA OS within the network that is being secured [25]. A network firewall analyzer likewise should be implemented to detect any anomalies in the network [11].

Another option to secure communications within a SCADA network is using a virtual private network (VPN). VPN offers a way of enabling specific individuals or user groups to establish on-demand data communications paths – secured using encryption protocols – to remotely access SCADA devices and networks as required. VPN technology also can help block attacks from malicious foreign entities by using geolocation services. Finally, IPsec is a framework of security standards to help secure communications sessions over IP networks by using encrypted keys and cryptographic security services. IPsec can be used in conjunction with VPN-secured SCADA communications to add an additional layer of security.

D. Standard Operating Procedures Recommendations

SOPs must be well-written and understood by all stakeholders working on the SCADA network. Creating a comprehensive security policy with training for clients, vendors, business partners, as well as regulatory agencies that have access to the network, is likewise essential. This policy should be a living document, which means always changing and updating when necessary.

Examples of security-related SOP provisions include: changing all factory default credentials; restricting administrator access to the control panel or certain IP addresses; enacting least privileges for all users with any ability to access the network; physically securing remote, unattended nodes; and establishing encryption guidelines and/or recommended standards for high-value, at-rest and in-transit data. Some of these SOPs are recognized as so important to ICS and critical infrastructure security that they should be elevated to become industry-wide standards, and a national-level certification process enacted to ensure they are followed.

SOPs also should include roles and responsibilities, and clearly state consequences for non-compliance to set policies. Prior to completing a security policy – and before each update – vulnerability assessments should be performed to identify any flaws or gaps in the system, to ensure a full understanding of the system architecture and where threats may lurk.

An important aspect of secure operations is training the workforce. SOPs are useless if the organization fails to educate all employees in the safe and secure behaviors the SOPs intend to support, both at work and on their home networks. Social engineering is one of the most frequently used ways to attack network infrastructures including SCADA, because the weakest security links remain the human ones. Social engineering comes in many forms – spoofing, fraudulent patch downloads, malware-bearing USB drives, and pretexting. Such techniques are difficult to detect and resist if they have not been anticipated and provided for in the SOPs. In addition, social engineering training should be performed on a regular basis to harden all employees with any means to access the network.

VII. Conclusion and Future Directions

This paper provided some historical perspective on SCADA technology and the pervasiveness of its associated risks. Vulnerabilities abound, partly due to how SCADA technology has not always evolved in step with emergent security threats and defensive solutions, while nonetheless continuing to promulgate into virtually every area of ICS and critical infrastructure. Informed by known general technology vulnerabilities and SCADA security gaps, a significant use case was analyzed. This suggested some way-ahead recommendations.

The evolving threat landscape means perfect system security can never be guaranteed. As new SCADA devices and systems become available, combining with older systems, and integrating newer technologies such as cloud and IoT, a "left-shift" is occurring in how security concerns are addressed. That is, there is renewed awareness that cybersecurity must be paramount from the earliest point of conceptualization, at each stage along the way to system deployment, and then continuously revisited throughout operations. This is a significant paradigm change for SCADA, but a necessary one to salvage such an important and critical technology.

As a means to assess vulnerabilities and to analyze threat vectors, using modeling and simulation (M&S) approaches is particularly challenging for SCADA since there are at least three attack categories to consider: known attacks for which reliable security countermeasures are known and implementable; known attacks against which a particular SCADA-enabled device/system/environment may not be readily defensible; and still other as yet unknown attacks. The latter case may be the most worrisome, since how secured a system is against unknown threats can only ever be speculated. In other words, how vulnerable SCADA networks are to unknown threats is unknown; but if that risk can be inferred based on the known threats, then it is considerable. As an additional future direction, platforms like CybatiWorks [35], Shodan [37], and Nessus should be leveraged as building blocks for more effective, M&S-based test beds for SCADA.

The precautionary measures recommended in this paper provide a roadmap for methodically filling in security gaps present in SCADA networks in four key areas: hardware, software, communications, and SOPs. Another follow on effort could involve building a scorecard for organizations to assess key elements within these areas in all their SCADA solutions. This can help identify those practices and components that pose the greatest risk to overall cybersecurity in their SCADA environments, and provide a rationalized means to prioritize revision, replacement or removal of problematic components and guidance.

Some experts have observed that while the established security prioritization for traditional systems is the confidentiality, integrity and availability (CIA) triad, the prioritization enacted for SCADA systems instead appears to be availability, integrity and confidentiality (AIC) [27]. An early design focus in SCADA on availability and ease of access, coupled with naïve reliance on security through obscurity, exposed SCADA systems to future compromises – and the future is now. It is in the national best interest to prioritize security in the SCADA-enabled systems used to light homes, to treat and distribute water, to enable financial transactions, and in so many other critical areas of industry and infrastructure.

References

[1] A. Ujvarosi, "Evolution of SCADA Systems," Bulletin of the Transilvania University of Bras¸ov, vol. 9, no. 58, pp. 63-68, 2016.

[2] H. Bentarzi, M. Tsebia and A. Abdelmoumene, "PMU Based SCADA Enhancement in Smart Power Grid," in 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018), Doha, Qatar, 2018.

[3] TrendMicro, "The State of SCADA HMI Vulnerabilities," https://www. trendmicro.com/vinfo/us/security/news/vulnerabilities-and-exploits/the-state-of-scada-hmi-vulnerabilities

[4] J. McCarthy and W. Mahoney, "SCADA Threats in the Modern Airport," International Journal of Cyber Warfare and Terrorism (IJCWT), p. 8, 2013.

[5] "Global ICS & IIoT Risk Report," https://cdn2.hubspot.net/hubfs/ 2479124/Report %20- %20Global %20ICS %20 & %20IIoT%20Risk %20Report.pdf

[6] K. Coffey, R. Smith, L. Maglaras and L. Maglaras, "Vulnerability Analysis of Network Scanning on SCADA Systems," Security and Communication Networks, vol. 2015, no. 5, 2018.

[7] G. Johnson, "Trends in Security Incidents in the SCADA and Process Industries: A Summary — Part 1," https://www.processonline.com.au/content/software-it/article/trends-

in-security-incidents-in-the-scada-and-process-industries-a-summary-part-1-11

[8] G. Johnson, "Trends in Security Incidents in the SCADA and Process Industries: A Summary — Part 2," https://www.processonline.com.au/content/software-it/article/trends-in-security-incidents-in-the-scada-and-process-industries-a-summary-part-2-60

[9] V. M. Igure, S. A. Laughter and S. A. Laughter, "Security Issues in SCADA Networks," Computers and Security, vol. 25, no. 7, pp. 498-506, 2006.

[10] O. Moreno, "SCADA," https://www.slideshare.net/orlandomoreno/scada-1964031

[11] C.-W. Ten, C.-C. Liu and G. Manimaran, "Vulnerability Assessment of Cybersecurityfor SCADA Systems," IEEE Transactions on Power Systems, vol. 23, no. 4, pp. 1836-1846, 2008.

[12] E. J. Byres, M. Franz and D. Miller, "The Use of Attack Trees in As sessing Vulnerabilities in SCADA Systems," in IEEE Conf. International Infrastructure Survivability Workshop (IISW '04), 2004.

[13] I. N. Fovino, M. Masera, L. Guidi and G. Carpi, "An Experimental Platform for Assessing SCADA Vulnerabilities and Countermeasures in Power Plants," in 3rd International Conference on Human System Interaction, Rzeszow, Poland, 2010.

[14] C. Wang, L. Fang and Y. Dai, "A Simulation Environment for SCADA Security Analysis and Assessment," in 2010 International Conference on Measuring Technology and Mechatronics Automation, Changsha City, China, 2010.

[15] "21 Steps to Improve Cyber Security of SCADA Networks," https://www.hsdl.org/?view&did=1826

[16] K.-b. Lee and Jong-in Lim, "The Reality and Response of Cyber Threats to Critical Infrastructure: A Case Study of the Cyber-terror Attack on the Korea Hydro & Nuclear Power Co., Ltd.," KSII Transactions on Internet and Information Systems, vol. 10, no. 2, 2016.

[17] "Background to "Assessing Russian Activities and Intentions in Recent US Elections": The Analytic Process and Cyber Incident Attribution," https://www.dni.gov/files/documents/ICA 2017 01.pdf

[18] "Department of Homeland Security, Critical Infrastructure Sectors," https://www.dhs.gov/cisa/critical-infrastructure-sectors.

[19] "SCADA Strange Love," http://www.scada.sl/

[20] S. Samtani, S. Yu, H. Zhu, M. Patton, J. Matherly and H. Chen, "Identifying SCADA Systems and Their Vulnerabilities on the Internet of Things: A Text-Mining Approach," IEEE Intelligent Systems, vol. 33, no. 2, pp. 63-73, 2018.

[21] E. Kovacs, "New SCADA Flaws Allow Ransomware, Other Attacks," https://www.securityweek.com/ new-scada-flaws-allow-ransomware-other-attacks

[22] N. Perlroth and C. Krauss, "A Cyberattack in Saudi Arabia Had a Deadly Goal. Experts Fear Another Try," https://www.nytimes.com/2018/03/15/ technology/saudi-arabia-hacks-cyberattacks.html

[23] D. Storm, "Hackers Exploit SCADA Holes to Take Full Control of Critical Infrastructure," https://www.computerworld.com/article/2475789/ hackers-exploit-scada-holes-to-take-full-control-of-critical-infrastructure. html

[24] D. Zahn and S. Hollis, "ICS Cybersecurity and the Devil's Rope," https://www.power-eng.com/articles/print/volume-122/issue-1/features/ ics-cybersecurity-and-the-devil-s-rope.html

[25] J. Gao, J. Liu, B. Rajan, R. Nori, B. Fu, Y. Xiao, W. Liang and C. L. P. Chen, "SCADA Communication and Security Issues," Security and Communication Networks, vol. 7, no. 1, pp. 175-194, 2014.

[26] H. A. Abbas, "Future SCADA Challenges and the Promising Solution: The Agent-Based SCADA," International Journal of Critical Infrastructures (IJCIS), vol. 10, no. 3/4, 2014.

[27] K. Stouffer, J. Falco and K. Kent, "NIST SP 800-82 Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security," NIST, 2006.

[28] Fortinet,"IndependentStudyPinpointsSignificant SCADA/ICS Cybersecurity Risks," https://www.fortinet.com/content/dam/fortinet/assets/white-papers/WP-Independent-Study-Pinpoints-Significant-Scada-ICS-Cybersecurity-Risks. pdf

[29] Dragos, "TRISIS Malware," https://dragos.com/wp-content/uploads/TRISIS-01.pdf

[30] "AnalyzingtheTRITONindustrialmalware," https://www.midnightbluelabs.com/blog/2018/1/16/analyzing-the-triton-industrial-malware.

[31] "Technical Product Guide Tricon Systems," US Nuclear Regulatory Commission, 2006.

[32] "Final Safety Evaluation for the Triconex Topical Report," US Nuclear Regulatory Commission.

[33] "Developer's GuideTriStation 1131," US Nuclear Regulatory Commission.

[34] "Vulnerability Spotlight: Multiple Vulnerabilities in Moxa EDR-810 Industrial Secure Router," https://blog.talosintelligence.com/2018/04/ vuln-moxa-edr-810.html

[35] CybatiWorks, https://cybati.org/

[36] Shodan, https://www.shodan.io

[37] Nessus, https://www.tenable.com/products/nessus-home