**A Study on Vulnerabilities and Threats on SCADA Devices**

Dawn Silverman, Yen-Hung (Frank) Hu, Mary Ann Hoppa

Department of Computer Science

Norfolk State University, Norfolk, Virginia

**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. To address our concern on this issue, this research conducts a thorough survey and investigation on cyber-attacks targeting SCADA systems adopted by most critical infrastructures and proposes solutions and recommendations for mitigating such attacks. This research first studies 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 is taken on the Schneider Triconex Tricon 3008 safety system as an instructive use case. Then, an assessment methodology is proposed along with some recommendations for methodically securing SCADA networks in general. 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 measurement and assessment.

**1. INTRODUCTION**

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

Critical infrastructure throughout the United States and abroad, including electric power grids and water purification and delivery, is dependent on SCADA devices to automate and monitor processes. Electric power grids supply power for residential and commercial property, where the loss of electricity could pose a significant challenge with dire consequences. As one example, residents in need of home health care may need machines that monitor their vital signs and sustain their quality of life, and losing electricity could have devastating effects on patient care.

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

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.

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 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 also pose another vulnerability issue. The previous version of SCADA systems were standalone, now these systems are also connected to systems not directly related to process control and monitoring. Not only does the effort to reduce the cost and improve performance, both control system vendors and owners have been transitioning from more proprietary technologies to the less expensive technologies prevalent in the IT world, such as Ethernet, TCP/IP, and Microsoft Windows [7] [8]. As additional components of control systems become more interconnected with the outside world, the probability and impact of a cyberattack will heighten [7] [8]. In fact, with the information age growing everything could be found online, even the flaws in SCADA specific technologies have become readily available to the public, which poses an even greater risk.

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 problem. Section 3 discusses known SCADA vulnerabilities and attacks. Section 4 discusses key SCADA vulnerabilities including hardware, software, communications and standard operating procedures. Section 5 introduces a real SCADA case – vulnerabilities in Schneider Triconex control system. Section 6 proposes potential resolutions to key SCADA vulnerability. Section 7 concludes the paper with a summary of suggestions for future work that builds upon these findings.

**2. 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 critical infrastructures has come under intense scrutiny. Moreno gives a quick overview of SCADA technology for those new to the topic in [9], while [10] explores crucial research issues involved in strengthening cybersecurity in SCADA networks. Ten, Liu and Manimaren [11] considered the problem through the lens of electric power systems. They proposed a framework for SCADA vulnerability assessment at three levels: system, scenarios and access points. Attack trees are another popular assessment methodology that has been applied to both SCADA specifications and deployments [12]. Coffey et al. [6] 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. This suggests that modeling and virtualization may be recommendable for building SCADA vulnerability assessment platforms similar to [13] and [14].

In 2001 President Bush created 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 SCADA networks. With that understanding of SCADA networks they were better able to make the necessary steps to secure them.

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 his agenda and therefore 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 cyber threats [16]. These executive orders directly affect the security of SCADA devices, since some areas of SCADA networks may be accessible through the internet. Moreover, the order to support an ongoing investigation into the United States 2016 election hack [17], and the order to secure critical infrastructure sectors [18] [5] are also important to secure SCADA systems.

A recent study [5] found the following ten vulnerabilities to be 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 Zero Day Initiative (ZDI) examined the state of SCADA HMI devices in particular, and reported in [3] that the most exploited vulnerabilities are:

* Memory corruption (20%)
* Credential management (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 SCADA vulnerabilities in industrial control systems (ICS) and programmable logic controllers, with five percent of these being dangerous remote code execution attack vectors.

As the Internet of Things (IoT) becomes pervasive in everyday life, more industrial systems and infrastructures are connecting to the internet. Thus, no segment can persist in thinking that SCADA risk is some other segment’s problem. The emerging fourth 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 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. In this light, the attack examples should come as no surprise [16] [22] [23] [24]. Revelations and events such as these have alerted industries, governments and even private citizens to the abundance and range of SCADA vulnerabilities, which all too often set a low bar for threat actor’s intent on disrupting critical infrastructure.

**4. Key SCADA Vulnerabilities**

In light of the prior general and specific examples of SCADA vulnerabilities, this section takes a more methodical look at SCADA vulnerabilities from four perspectives: hardware, software, communications and standard operating procedures (SOPs).

**4.1 Hardware Vulnerabilities**

Hardware vulnerabilities cover such items as RTUs, HMI, programmable logic controller (PLC), and smart devices that report back to the main terminal in a SCADA system.

The typical SCADA environment use a HMI which all smart devices are monitored by providing information to the operators. Information to the operators are state of the control systems or specific sensors, and also provides a means to facilitate any corrective measures that may need to be taken. According to [3] HMIs are often the source of a primary target within a SCADA system, and they suggest HMIs should be installed on an air-gapped or isolated on a trusted network due to the vulnerabilities they pose. In today’s SCADA networks these HMIs are highly advanced and can be customized to monitor a current state of a system.

The RTUs within a SCADA system are usually centrally controlled by the master system. These RTUs usually consist of devices such as relays, actuators, circuit power breakers, voltage regulators, and a multitude of sensors that are attached to say the least. Typical SCADA environments are interconnected to the master station through a variety of channels which pose their own vulnerabilities to the RTUs. They are sometimes interconnected through radio links, leased lines, fiber optics and others [25].

PLCs within a SCADA environment pose their own set of vulnerabilities according to the research presented in [26]. They have limited capability for implementing advanced control algorithms. In a critical infrastructure vulnerability test performed by [20] which tested 1,843 devices and out of 1,207 there were critical vulnerabilities that were found in Allen-Bradley Micro Logix 1400 PLC. According to that study attackers can use buffer overflow and man-in-the-middle attacks that harm the device and the infrastructure it controls [20].

**4.2 Software Vulnerabilities**

Software vulnerabilities in a SCADA environment include certain operating systems used for SCADA systems. Most SCADA systems tend to use common operating systems such as UNIX, Solaris, or Windows in the control center systems [25]. 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. The statistics for software attacks are maintained by the Computer Emergency Response Team/Coordination Center (CERT/CC) and the US-CERT.

The statistics from these organizations show that software technology over decades has significantly increased the number of known operating systems vulnerabilities and security holes. These statistics are not complete due primarily because organizations are reluctant to publicly disclose the statistical dataset about intrusion attempts [11].

Another vulnerability for SCADA software involves multitasking and real-time databases that are implemented in many SCADA environments. They implement several servers that are responsible for data acquisition and handling many parameters which are typically those to which they are connected [25]. Many SCADA systems adopt less expensive Microsoft Window systems for financial reasons, which have a broad variety of vulnerabilities. The use of embedded operating systems in SCADA results in many additional expenditures because it is tougher to interact with these systems. This is one of the main reasons SCADA systems tend to use the more common operating systems such as Microsoft Window, UNIX, and Solaris in the control center systems [25].

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

* Viruses, malware, and Trojan horses: Opening attachments or clicking on links from unknown or spoofed emails. Downloading software updates or patches could contain malicious content. Not patching software when they become available leaves a system open for attack. Security holes that are not yet known to the software developers are known as a Zero-Day vulnerability.
* Logical errors: They are generated during the code writing of the system and may cause unintended or undesired output. Logical errors may not be immediately known to the user until an unexpected or incorrect result is produced.
* Convenient features for users: They are infections, such as file downloading, from features used by most users. A great example of this would be internet e-mail based on SMTP which was developed for convenience. This standard e-mail system suffers numerous vulnerabilities that may engage users, such as e-mail spoofing, eavesdropping, and malicious content.
* Reconfiguring the authentication permissions: Administrators should always practice adding higher level authentication standards for SCADA systems.
* Administrator access: Restricting administrator access to control panel or certain IP addresses may be one of the most effective way to mitigate malicious attacks. Restricting administrative access and privileges may also prevent targeted cyber-attacks that use escalation into a SCADA system. Least privileges for all users of SCADA systems is an effective way to allow employees to do their job without unnecessary rights to areas they do not need.

**4.3 Communications Vulnerabilities**

A typical SCADA communication system consists of a master station and many other distributed RTUs. These RTUs are interconnected to master stations through a variety of communication channels and protocols [25]. One of the greatest challenges is that the channel limits the speed of data acquisition control that can be performed. Furthermore, random noise on the channel is another challenge that has hindered SCADA communication. The interconnection of microprocessors used in SCADA has been an increasing trend, and this interconnection is what makes SCADA systems less secure [25].

The new generation of SCADA devices can now be accessed from anywhere in the world, legacy SCADA was implemented back in the 1970’s and were isolated. Some of the legacy devices are still in use today and are not upgraded or patched for today’s internet challenges. Communication between these devices leaves them open for attacks such as the Man-in-the-Middle attack, which is when an attack reroutes communication so they may gain control. Denial of service is another attack, which may severely diminish how this hardware operates and the amount of time it reports back to the main unit of the SCADA devices.

**4.4 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 measures are both consistent and current to protect against malicious attacks. According to NIST, SOPs should focus on systems holistically rather than just individual devices, and should include PLCs, DSCs, SCADA, and instrument-based systems that use a monitoring device such as HMI [27].

If SOPs are not regularly re-evaluated, they may not include the most up-to-date information to secure 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. Overlooking such details provides avenues for attackers to penetrate into the SCADA environment. Other oversights may include lax identity management and administrator accounts with well-known default configurations that are easy to hack. An absence of encryption protocols for master and slave device communications, and no evidence 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].

**5 SCADA Vulnerability Use Case**

To identify additional areas of SCADA that should be scrutinized more closely, it can be useful to analyze 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 plan shut down. This section will analyze information from various published case studies for hardware vulnerabilities, software vulnerabilities, and communication vulnerability related to the Schneider Triconex 3008 that was hacked. Walking through this incident show that simply following the manufacturer’s instructions could have prevented this breach from occurring.

**5.1 Schneider Triconex 3008 Safety Controller**

The safety system attacked in this case consisted of the Schneider Triconex 3008 running the TriStation 1131 software. The Trisis, Triton, or HatMan, malware used in the attack was developed to replace logic in a final control safety element in a SCADA environment. This malware targeted the Triconex Tricon safety shutdown system. The researchers at Dragos have laid out alternative architectures and explanations of each suggestion they have for safeguarding these safety security controllers [29].

Triton is the first publicly known examples of malware that specifically targets Industrial Control Systems. Typically, these systems have limited connectivity and are segregated from the rest of the Operational Technology (OT), so to limit the potential for malicious attack [30].

**5.2 Vulnerabilities in Triconex Hardware**

The Schneider Triconex 3008 is a safety control system, main processor that provide 16 megabytes of DRAM, which is used for the control program, sequence-of-events data, I/O data, diagnostics and communication 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 on one power module or power rail will not affect the performance of the system [31].

Triconex controller is working on a solution to this malware and the U.S. Department of Homeland Security is heading an investigation into this matter. Schneider Electric states that this legacy Tricon system was working properly and the hack could not have been implemented if the keyswitch had been set to the proper mode. Specifically, the controller should have been set to the “run mode” which prohibits any logic changes. Instead, it has been left in the “program mode.” The specific malware that was installed had the capability to scan these controllers and issue commands to them. An additional vulnerability that was identified is that all Tricon Controllers are shipped with identical keys, and there is currently no procedure in place for a customer to order a different key for their systems [32].

**5.3 Vulnerabilities in Triconex Software**

Triton malware gained remote access to a Triconex engineering workstation running Microsoft Windows as well as Distributed Control System (DCS). “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]. This malware is said to reprogram the safety controller using the TriStation software that is used to run the system. TriStation software protocol is proprietary and undocumented, which means that the attacker had to reverse engineer it, possibly through a combination of using similarities with the documented Triconex System Access Application (TSAA) protocol [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 ia 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 but apparently no such payload was recovered during the incident since the attacker was discovered while preparing the implant of this malware” [30].

The TriStation protocol is typically set up as UDP-based serial over Ethernet, which is the standard for the industrial control world. The request packets contain a 2-byte function code (FC), which is then followed by a counter ID, length field and a request data together with checksums. The Triton framework attack lies in the following sequence of function codes and expected response codes [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 creates a user name and password with the highest level of privileges, which is Level 01. However, by default, the user name is “MANAGER” and the default password is “PASSWORD” [33]. This information TriStation guide is posted online and available to anyone, this poses a major vulnerability if these passwords are not changed. Many times default usernames and passwords are kept so they are easy to manage and available for all the users to get into. A password may also be required for connecting to the controllers themselves, but not present initially. The TriStation protocol itself is unencrypted, therefore any attacker that is capable of observing network traffic between the controller and the TriStation workstation is likely to be able to circumvent such a protection [30].

**5.4 Vulnerabilities in Triconex Communication**

The Triconex industrial safety controller has many different communication modules to facilitate serial and network communications across a variety of protocols. One example is the Tricon Communication Module (TCM) which allows communication 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]. The Modbus protocol was used to analyze a real-time vulnerability in [34] because of the following reasons.

* 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.

In the case study presented in [34] Wireshark was utilized to analyze the network traffic. They performed two well-known attacks in a test bed configured using Modbus, which were the denial-of-service attack (DoS), and the man in the middle (MITM) attack. They analyzed that both attacks resulted in a severe impact on system operation and stability. They also realized the fact that Modbus protocol is too trusting in nature with no access control lists and no form of trust domain [34].

**6. Recommendations to Key SCADA Vulnerabilities**

Due to their importance to critical infrastructure and quality of life, SCADA networks must be safeguarded. While there is no such thing as a perfectly secure system, this research has uncovered a number of precautions and mitigations that can help alleviate some vulnerability and thereby hinder the fulfillment of malicious actions against SCADA. These recommendations are grouped into four areas common to all SCADA systems, where specific actions can be undertaken to harden them against cyber-attacks: hardware, software, communications, and standard operating procedures.

These recommendations provide a roadmap for a methodical approach to filling the security gaps present in SCADA networks. One follow-on effort could involve building a scorecard that organizations can use to assess key elements within the hardware, software, communications and SOPs of their overall SCADA solutions, identifying those components that pose the greatest risk to overall cybersecurity, and prioritizing their repair or replacement.

**6.1 Hardware Recommendations**

In today’s SCADA environment there are still many legacy devices in use. Legacy devices that can no longer be upgraded or patched for vulnerabilities should be 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 system. SCADA hardware devices always should have physical controls limiting who can access them, and if they are in a remote location they should be monitored and locked.

**6.2 Software Recommendations**

The most critical software recommendation is keeping the software used on SCADA networks up-to-date and patched. New versions and patches are generally released to improve security features and functionality; 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 authentic providers’ websites and not spoofed websites trying to penetrate the network.

Antivirus software on any network reduces the possibility of malicious content from causing harm to devices. Because SCADA device are typically used in our nations critical infrastructure they in particular should be configured using the best antivirus software. Antivirus software should be updated often to protect against the latest malware trying to penetrate a SCADA network.

The use of an embedded operating system in a SCADA network decreases the likelihood of attack 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 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.

**6.3 Communications Recommendations**

Firewalls are a defense mechanism that determines the packet flow between two networks. There may be several different security trust levels between networks, firewalls are configured to filter out unnecessary traffic in either direction. Firewall criteria to allow incoming or outgoing traffic should be configured for types of protocols, what traffic may come in or out of a network, specific port services or certain port service range, as well as specific IP addresses or range of addresses [11]. A firewall can be implemented either by separate hardware connected externally to the network or by software integrated into the operating system of SCADA, which is being secured in the network [25]. Due to the sensitivity of SCADA networks, they should be monitored to have a baseline of acceptable use or traffic. A network firewall analyzer should be implemented to detect any anomalies running in the network [11].

One of the most successful ways to help secure communications within your SCADA network is using a virtual private network (VPN) that allows a more secure way of connecting to a remote SCADA network. The use of VPN networks will allow data paths to be more secret to a certain extent and allow users or groups to access devices more securely. The VPN networks should always use encrypted protocols for communication between devices. The use of VPN technology can also help block malicious attacks from outside of the country using a geolocation services. Another protocol used with VPN technology is IPsec which is a framework of security standards to help secure communication over internet protocol (IP) networks. The use of IPsec uses cryptographic security services which uses encrypted keys during sessions of communication.

**6.4 Standard Operating Procedures Recommendations**

SOPs must be well written and understood by all employees working on the SCADA network. Creating a comprehensive security policy with training for all employees, 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. SOPs should include roles and responsibilities of all employees, and clearly state consequences for non-compliance to set policies. Prior to completing a security policy – and before each update – vulnerability assessments must be performed to identify any flaws or gaps in the system and to ensure a full understanding of architecture and where threats may exist. Social engineering training should be performed on a regular basis for all employees with any means to access the network.

An important aspect of secure operations is training the workforce. Organizations often fail to educate all employees in safe and secure behavior, 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 in any network the weakest links are the human ones. Social engineering comes in many forms – spoofing, patch downloads, malware-bearing USB drives, pretexting. Such techniques are difficult to detect and resist if they have not been anticipated and provided for in the SOPs.

**7. Conclusion**

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 critical infrastructure. Several use cases were examined, informed by known SCADA security gaps and best practices from parallel disciplines such as physical security. Some potential vulnerability assessment (VA) approaches grounded in existing technical platforms and methods were proposed, as well as a power-grid specific solution.

The evolving threat landscape means one hundred percent 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 this important and critical technology. As additional way-ahead recommendation, platforms like CybatiWorks Shodan, and Nessus should be leveraged as building blocks for more effective, M&S-based vulnerability assessment platforms for SCADA.

Modeling devices and cyber-attacks as a means to assess and mitigate vulnerabilities is particularly challenging since there are at least three attack categories to consider for SCADA: 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.

Some experts have observed that while the security triad prioritization in traditional systems is confidentiality, integrity and availability (CIA), a more typical prioritization enacted for SCADA systems 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 our nation’s best interest to reprioritize security in the SCADA-enabled systems we use to light our homes, to treat and distribute the water we drink, to enable our financial transactions, and in so many other critical areas.

# References

|  |  |
| --- | --- |
| [1] | A. Ujvarosi, "Evolution of SCADA Systems," *Bulletin of the Transilvania University of Braşov,* 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," 23 5 2017. [Online]. Available: 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] | www.cyberx-labs.com, "Global ICS & IIoT Risk Report," 10 2017. [Online]. Available: 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," 20 9 2007. [Online]. Available: https://www.processonline.com.au/content/software-it/article/trends-in-security-incidents-in-the-scada-and-process-industries-a-summary-part-1-1197351337. |
| [8] | G. Johnson, "Trends in Security Incidents in the SCADA and Process Industries: a Summary — Part 2," 20 10 2007. [Online]. Available: https://www.processonline.com.au/content/software-it/article/trends-in-security-incidents-in-the-scada-and-process-industries-a-summary-part-2-605023384. |
| [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," 7 9 2009. [Online]. Available: 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 Assessing 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," 2002. [Online]. Available: 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," 6 1 2017. [Online]. Available: https://www.dni.gov/files/documents/ICA\_2017\_01.pdf. |
| [18] | "Critical Infrastructure Sectors," [Online]. |
| [19] | "SCADA StrangeLove," [Online]. Available: 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," 27 4 2017. [Online]. Available: 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.," 15 3 2018. [Online]. Available: 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," 15 1 2014. [Online]. Available: 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," 1 1 2018. [Online]. Available: 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, "Independent Study Pinpoints Significant SCADA/ICS Cybersecurity Risks," [Online]. Available: https://www.fortinet.com/content/dam/fortinet/assets/white-papers/WP-Independent-Study-Pinpoints-Significant-Scada-ICS-Cybersecurity-Risks.pdf. |
| [29] | Dragos, "TRISIS Malware," [Online]. Available: https://dragos.com/wp-content/uploads/TRISIS-01.pdf. |
| [30] | "Analyzing the TRITON industrial malware," 16 1 2018. [Online]. Available: https://www.midnightbluelabs.com/blog/2018/1/16/analyzing-the-triton-industrial-malware. |
| [31] | "Technical Product Guide Tricon Systems," US Nuclear Regulatiory 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," 13 4 2018. [Online]. Available: https://blog.talosintelligence.com/2018/04/vuln-moxa-edr-810.html. |