Identifying SCADA Vulnerabilities Using Passive and Active Vulnerability Assessment Techniques

Sagar Samtani, Shuo Yu, Hongyi Zhu, Mark Patton, Hsinchun Chen
Management Information Systems
The University of Arizona
Tucson, AZ 85721
{sagars, shuoyu, zhuhy, mpatton}@email.arizona.edu, hchen@eller.arizona.edu

***Abstract*— Critical infrastructure such as power plants, oil refineries, and sewage are at the core of modern society. Supervisory Control and Data Acquisition (SCADA) systems were designed to allow human operators supervise, maintain, and control critical infrastructure. Recent years has seen an increase in connectivity of SCADA systems to the Internet. While this connectivity provides an increased level of convenience, it also increases their susceptibility to cyber-attacks. Given the potentially severe ramifications of exploiting SCADA systems, the purpose of this study is to utilize passive and active vulnerability assessment techniques to identify the vulnerabilities of Internet enabled SCADA systems. Specifically, we collect a large testbed of SCADA devices from Shodan, a search engine for the IoT, and assess their vulnerabilities with Nessus and against the National Vulnerability Database (NVD). Results of this study indicate that many SCADA systems from major vendors such as Rockwell Automation and Siemens are vulnerable to default credential, man-in-the-middle, and SSH exploit attacks.**

Keywords—SCADA; Shodan; vulnerability; passive vulnerability assessment; active vulnerability assessment; Nessus; National Vulnerability Database

I. INTRODUCTION

Modern society is reliant on industrial and critical infrastructure such as power plants, oil refineries, sewage, and transportation services. To help ensure their effective and efficient operations, Supervisory Control and Data Acquisition (SCADA) systems were designed in the 1960's to allow human operators to supervise, maintain, control, and collect data from critical infrastructure [1]. Today, many popular SCADA manufacturers such as Siemens and Rockwell Automation incorporate networking technology (e.g., TCP/IP protocols) to allow operators to remotely control SCADA systems via the Internet.

While connecting SCADA systems to the Internet provides an increased level of convenience, it also exposes them to a variety of traditional cyber-attacks such as buffer overflow, memory, and Denial of Service (DoS) attacks [1]. Additionally, tools such as Shodan, a search engine for the Internet of Things (IoT), regularly scan index, and publicly provide information about accessible SCADA systems [2]. Figure 1 illustrates how a user can use a SCADA specific query on Shodan to retrieve information about SCADA systems. Figure 2 further illustrates how a user can utilize those results to access SCADA system interfaces.

Malicious actors could cause societal harm by exploiting the publicly accessible SCADA systems. Such concerns have led world leaders such as President Obama to state that we must "protect our nation's most sensitive infrastructure from cybersecurity threats" [3]. Given these motivations, the purpose of this research is to identify SCADA system vulnerabilities. Specifically, we adopt scalable passive and active vulnerability assessment techniques on a large testbed of SCADA devices identified from the Shodan database.

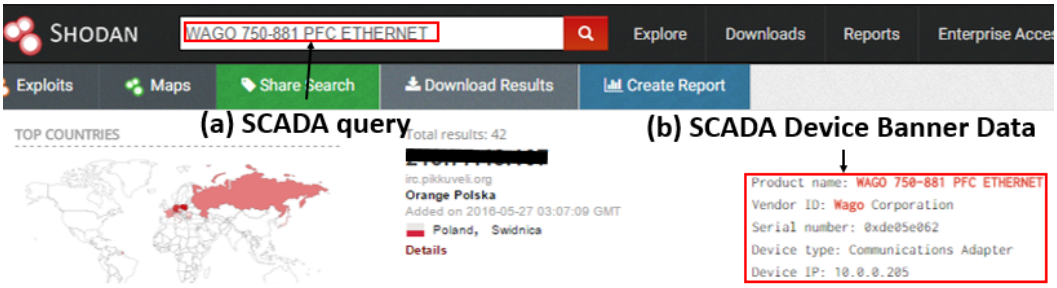

Fig. 1. Shodan Results for a SCADA specific query (WAGO 750-881 PFC Ethernet)

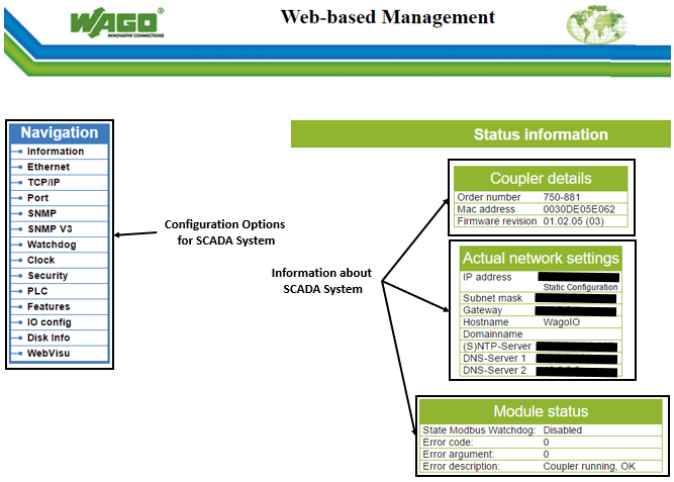

Fig. 2. Web Based Control Panel of a PFC Ethernet SCADA system

The remainder of this paper is organized as follows. We first review literature on modern SCADA systems, their security concerns, and vulnerability assessment techniques. Subsequently, we detail our research testbed and design. We then summarize our key findings and results. Finally, we highlight several promising directions for future research and conclude this research.

II. Literature Review

To form the basis for this research, we review two major areas of literature: 1) SCADA background and security to gain deeper insight into the different components of SCADA systems and their associated security concerns, and 2) vulnerability assessment techniques to understand state-of-the art methods to assess the vulnerabilities of systems.

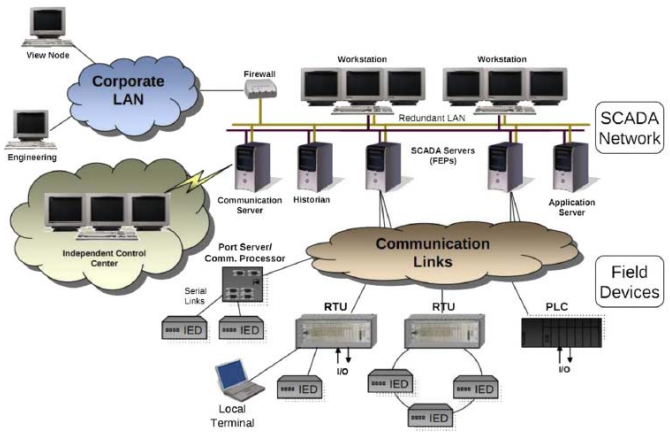

Fig. 3. Conceptual Industrial Network with SCADA Systems [1]

A. SCADA Background and Security Concerns

Networks with SCADA systems generally have three components: a corporate network containing traditional workstations and servers; a SCADA network comprising of Human Machine Interfaces (HMI's), SCADA servers, and Historians which log collected data; and a field device network, containing Programmable Logic Controllers (PLC's), Remote Terminal Units (RTU's), Master Terminal Units (MTU's) and Intelligent Electronic Devices (IED's). Operators use the HMI's at the SCADA network level to send commands to the field devices to adjust settings or parameters for the critical infrastructure (e.g., adjust the oil level in a refinery). Field devices such as PLC's, RTU's, or MTU's take the input from either HMI's or the critical infrastructure itself and maintain the infrastructure accordingly. These devices typically use specialized or proprietary protocols such as Modbus, Common Industrial Protocol (CIP), or Distributed Network Protocol (DNP3) to communicate with each other [1]. Figure 3 illustrates an example of an industrial network with SCADA components.

For the majority of their existence, SCADA systems and networks have remained isolated from the open Internet. As a result, SCADA security concerns primarily revolved around insider threats [4][5]. However, the use of TCP/IP protocols in modern SCADA systems has led to a heightened susceptibility to traditional exploits such as Operating System (OS) attacks and DDoS [6][7]. SCADA system Internet connectivity has also resulted in the exposure of SCADA specific protocols such as Modbus, CIP, and DNP3, many of which were never designed to be accessible via the Internet.

Modern SCADA systems often provide a web interface to operators for remote control. While convenient, these interfaces can be vulnerable to common web based attacks including SQL injection, Cross-Site Scripting (XSS), clear text submission of passwords, HTML injection, and directory traversal [1][8]. These attacks have the potential to exploit both the web interface and the SCADA system itself. Furthermore, these exploits can provide users with the ability to remotely control the entire system [9]. Studies recognize that outdated or unpatched software at the OS and application levels pose a great risk for SCADA systems [1].

Despite the plethora of literature highlighting potential SCADA vulnerabilities, we were only able to find one study identifying Internet enabled SCADA system vulnerabilities [2]. However, those authors only utilized password testing techniques to assess vulnerabilities, and did not address any of the other vulnerability concerns highlighted in SCADA security literature. Given our goal in assessing the breadth of vulnerabilities of SCADA systems on the Internet, we review techniques and tools commonly used in vulnerability assessment.

B. Vulnerability Assessment Techniques

Vulnerability assessment has traditionally been motivated by an organization's desire to identify vulnerabilities within their system and mitigate them prior to exploitation. Penetration testing literature defines a vulnerability as "a flaw within a system, application, or service which allows an attacker to circumvent security controls and manipulate systems in ways the developer never intended" [9]. Today, there are two widely accepted methods of vulnerability assessment: passive and active.

Passive vulnerability assessment techniques aim to cross-reference system specific characteristics (e.g., OS version, specific software versions, etc.) with databases of known vulnerabilities [9]. The National Vulnerability Database (NVD), managed by NIST, is a widely used database containing millions of records about specific device vulnerabilities. It is estimated that only 14% of software vulnerabilities disclosed in NVD are patched immediately after their release while 50% remain vulnerable after three months, and 30% remain vulnerable after six months [10]. Sadly, SCADA systems see an even lower patch rate (10%) compared to standard computing technology [11]. Such statistics have made NVD cross-referencing a viable form of passive vulnerability assessment. Recent years have seen an increased number of studies utilizing NVD cross-referencing to identify vulnerabilities of devices found on Shodan. For example, [12] created ShoVAT, an automated tool that discovered 4,000 potential vulnerabilities for 886 non-SCADA systems by NVD cross-referencing. Inherent in its low level of intrusiveness, passive assessment does not actively check to see if identified vulnerabilities truly exist within the systems.

In contrast to passive vulnerability assessment, active vulnerability assessment techniques actively probes devices to identify vulnerabilities. Examples of such assessments include port scanning, checking for SQL injections and HTML injections, attempting password logins, monitoring network traffic, and dropping malicious or exploitative payloads [9]. Unlike passive testing, various open source and enterprise tools are available for active assessment. These tools generally focus on specific tasks. For example, NMap and ZMap are used for port scanning, while Burpsuite and SQLMap are designed to probe web applications and databases, respectively [9].

Despite the focused nature of the majority of these tools, Nessus (maintained by Tenable Network Security) is viewed as a gold-standard for active vulnerability assessment [9]. Nessus is a robust enterprise level software designed to test a breadth of vulnerabilities for large networks. Unlike other vulnerability assessment tools, Nessus provides users with the ability to configure over 80,000 plugins, each of which is designed to assess a specific vulnerability. For example, Nessus can scan ports, identify web application vulnerabilities, discover unpatched operating systems and software versions, attempt default usernames and passwords in login fields, determine if systems have backdoors, and pinpoint database vulnerabilities. Nessus can also test for SCADA specific vulnerabilities including buffer overflow, Modbus coil exposure, and discovery of unsupported versions of SCADA software. Nessus categorizes each vulnerability into different thresholds of risk based on the Common Vulnerability Scoring System (CVSS) score. CVSS is an open industry standard for assess the severity of system security vulnerabilities. Table 1 summarizes each threshold and some sample vulnerabilities associated at each level.

Despite well-established passive and active vulnerability assessment techniques, tools, and approaches, we were unable to find any study using either technique to comprehensively assess the vulnerability of SCADA systems on the Internet of Things. Instead, the majority of studies focus on vulnerability assessment within specific networks [9]. Nevertheless, passive and active vulnerability assessment techniques have proven to be relatively scalable and useful in identifying vulnerabilities in large networks [9].

TABLE I. RISK THRESHOLDS FOR VULNERABILITIES IN NESSUS

Risk Level	CVSS Range	Examples of vulnerabilities
Critical	10.0	Default Credentials, Unsupported Unix operating system
High	7.0 – 9.9	SQL Injections, OpenSSH vulnerabilities, Buffer Overflows
Medium	4.0 – 6.9	DoS, XSS, Browsable Web Directories, Modbus coil access
Low	0.1 – 3.9	Cleartext submission of credentials, authentication without HTTPS
None (Informational)	0.0	Generic paramater injections, web mirroring

C. *Research Gaps and Questions*

We identified several research gaps from our literature review. SCADA literature has extensively documented the potential vulnerabilities of modern SCADA systems, but only one study has assessed vulnerabilities of Internet enabled SCADA devices on Shodan by testing default usernames and passwords over the Telnet protocol [2]. Despite comprehensive passive and active vulnerability assessment techniques and tools, no study has applied these approaches to assessing the vulnerability of Internet enabled SCADA systems. Based on these gaps, the following research questions are posed for study:

- How can active and passive vulnerability assessment techniques be used to identify Internet enabled SCADA system vulnerabilities?
- What types of vulnerabilities are Internet enabled SCADA devices susceptible to?

III. RESEARCH TESTBED AND DESIGN

We gathered information about 20,461 SCADA devices from the Shodan API. While we recognize that the Shodan API provides a subset of SCADA systems actually in Shodan's multi-billion record dataset, identifying additional SCADA devices without using the API is a non-trivial technical task requiring extensive text processing techniques. As such, we save identification of all SCADA devices in Shodan for future work. Nevertheless, our collection contains devices from major vendors such as Siemens, Rockwell Automation, Schneider Electric, and Allen-Bradley. For each device, Shodan provides details on scanned ports open on the machine. Overall, devices in our collection had open web specific ports (e.g., 80, 8080, 443, etc.), SCADA specific ports running SCADA protocols (e.g., 502, 44818, 47808, 1028, 102, etc.), and ports for general services such as Telnet and FTP. Shodan provides banner data generated from each of the ports (e.g., Telnet welcome message, Modbus slave information, etc.).

Given the characteristics of our dataset, we carefully crafted three major components in our research design; a

passive assessment cross-referencing Shodan banner data with the NVD, an active assessment utilizing Nessus on the SCADA devices, and a manual evaluation of the passive and active assessment results. Details of each component are depicted in Figure 4 and described in the following subsections.

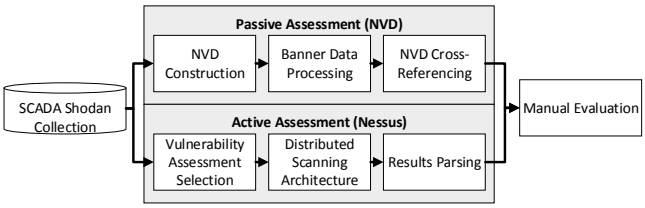

Fig. 4. Passive and Active Vulnerability Assessment Research Design

A. *Passive Vulnerability Assessment (NVD)*

To conduct passive vulnerability assessment, we first downloaded the NVD onto local disks. In total, there were 1,903,365 vulnerabilities posted since 1999. We then developed a Python script to automatically extract and cross-reference the vendor, product, and version number information provided the banner data with NVD records. Figure 5 and Table 2 show an example of how SCADA devices' banner data is cross-referenced with an NVD record.

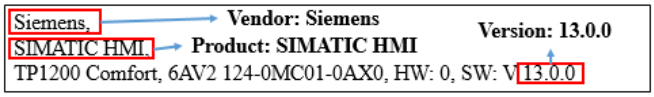

Fig. 5. Vendor, product, and version number matching from a Shodan banner

TABLE II. SAMPLE NVD ENTRY MATCHING TO THE VENDOR, PRODUCT, AND VERSION NUMBER FOUND IN FIGURE 5

Vendor	Product	Version Number	Vulnerability ID
Siemens	Simatic HMI	13.0.0	CVE-2015-2823

B. *Active Vulnerability Assessment (Nessus)*

The active vulnerability assessment component of our framework utilizes Nessus to actively probe the SCADA devices in our testbed. We selected Nessus over other active vulnerability assessment tools such as Burpsuite or NMap for several reasons. First, Nessus is a scalable, flexible, and well-maintained enterprise level software designed for testing a large amount of devices. Second, Nessus provides a rich set of plugins that can be configured to assess a multitude of vulnerabilities, ideal for a testbed containing a variety of devices. Finally, Nessus provides risk rankings and descriptions for each identified vulnerability.

Rather than taking a brute-force approach and testing every device for all vulnerability scans provided by Nessus, we adhere to penetration testing fundamentals by carefully selecting a subset of scans to use [9]. Such an approach is beneficial for two reasons. First and most importantly, limiting the number and types of scans reduces the potential for causing unintentional harm to the target systems. This is especially important for SCADA systems, as disrupting these systems may damage the critical infrastructure the system is controlling. Second, reducing the types of scans conducted also helps ensure a timely completion of the vulnerability assessment. Based on this rationale, the characteristics of our dataset, and prior literature, we set up Nessus to scan primarily for SCADA and web specific vulnerabilities, identify if devices were running unpatched OS or software versions, and discover their susceptibility to buffer overflow, DoS, and memory leaks. Additionally, we leveraged Nessus' built-in device recognition and username and password database to attempt default username and password credentials. We avoided port scanning, as such an activity has the potential to harm devices [9].

Given the relatively large amount of SCADA devices and the high resource consumption of Nessus, we created a distributed master-slave architecture of nine machines. Eight of the machines (slaves) was responsible for scanning a subset of devices in our collection. Each of these machines had 80 GB of SSD space, 32 GB of RAM, and 36 CPU's. The master machine (500 GB SSD, 16 GB RAM, 8 CPU's) was responsible for monitoring and collecting data from each of the slave machines. Overall, the entire scanning process took approximately 48 hours to complete. After scanning, all results were parsed into a database for further analysis.

IV. RESULTS AND DISCUSSION

Both the passive and active assessment identified a variety of vulnerabilities. The following subsections highlight the major vulnerabilities identified, the number of devices affected, and the major vendors for those devices. Although we attempted to identify the device owners by cross-referencing the IP address of vulnerable devices with GeoIP databases, over 90% of the results pointed to Internet Service Providers (ISP's) which the SCADA devices were operating on. As a result, such information provided us with inaccurate owner details and was excluded from our discussion.

A. *Passive Assessment (NVD)*

Passive assessment showed that 1,707/20,461 (8.34%) devices contained information which had a listed vulnerability in NVD. While our scripts identified a total of 123 different types of vulnerabilities, the majority (1,172/1,707) of the devices were vulnerable to one of three NVD entries. First, 975 devices are potentially vulnerable to an exploit which allows attackers to recover plaintext data from users who have logged into the SCADA system remotely. This vulnerability indicates that an attacker can essentially "spy" on users and log their data, similar to how a keylogger operates. Second, 197 devices are vulnerable to Denial of Service (DoS) attacks. This type of attack could cause significant damage as SCADA systems often require continuous uptime for critical infrastructure operation. The final vulnerability allows attackers to identify valid usernames for SCADA systems. Utilizing valid username and password combinations is often referred to as "the easiest and most effective way to gain access over a system" [9]. Discovering valid usernames for SCADA systems would allow attackers to refine their password cracking techniques, thus allowing them to gain control over the system.

B. *Active Assessment (Nessus)*

The active assessment results showed that 4,009/20,461 (19.59%) of the SCADA devices in our collection have 'Critical', 'High', 'Medium', or 'Low' risks. Specifically, 182 devices have 'Critical' risks, 189 have 'High' risks, 2,737 have 'Medium' risks, and 901 have 'Low' risks. An additional 5,101 devices returned informational responses to our scans. It should be noted, however, that Nessus was designed to be an internal network scanning tool. Given that the devices scanned are SCADA and accessible on the open Internet, the risks associated with these vulnerabilities may significantly higher be far more severe than what is indicated by Nessus. Table 3 breaks down selected vulnerabilities at each risk level, the number of devices affected for each vulnerability, and the major vendors affected by the vulnerability. For space considerations, we list the vulnerabilities at the 'Critical', 'High', and 'Medium' levels.

Table 3 reveals some interesting vulnerabilities at each risk level. The majority (131/182) of the 'Critical' vulnerabilities are due to the use of Default Credentials on MicroLogix 1400 Programmable Logic Controllers (PLC's) designed by Rockwell Automation/Allen-Bradley. PLC's are digital computers designed to automate and monitor the state of a variety of electromechanical processes such as electric motors, hydraulic cylinders, relays etc. Such technologies are often used in factory assembly lines, light fixtures, and industrial level heating and cooling units [13]. While it is difficult for us to identify what these PLC's control without analyzing their networks, Figure 6 illustrates how a user can leverage the default credentials to potentially change the PLC's settings. Adjusting these settings can potentially harm both the PLC itself and the infrastructure which it is controlling. To ensure device safety, we only highlight areas which users can adjust settings. We do not make any changes to the system.

In addition to the Rockwell Automation PLC's being susceptible to Default Credentials, Nessus also identified 14 HP and RuggedCom had storage and switch (respectively) SCADA technologies which could be accessed using Default Credentials. The use of Default Credentials indicates that an attacker may be able to exploit SCADA systems. Nessus also identified four Siemens Simatic HMI's and PLC's exploited with the Conficker Worm, one of the most popular and dangerous malware known to date. This worm is designed to steal sensitive information from machines [14]. Given that SCADA systems often contain critical information about their operations, such a worm could have significant consequences for an infected system.

TABLE III. SELECTED VULNERABILITIES AND AFFECTED VENDORS AT VARYING RISK LEVELS

Risk Level	Number of Devices	Vulnerability Name(s)	Selected Affected Vendors
Critical	131	Rockwell Automation MicroLogix 1400 PLC Default Credentials	Rockwell Automation/ABB
	15	InduSoft Arbitrary Script Execution	InduSoft
	14	Default Credentials	HP, RuggedCom
	4	Conficker Worm Detection	Siemens
High	111	OpenSSH and DropBear SSH Vulnerabilities	Rockwell Automation/ABB, Siemens, Schneider Electric, Honeywell
	29	Default Credentials	Schneider Electric
Medium	1,407	Unencrypted Telnet Server	Rockwell Automation/ABB, Siemens, Schneider Electric, Power Measurement, Acromag, Honeywell
	607	Modbus Coil Access	Schneider Electric, Rockwell Automation/ABB, Acromag, Lantronix, Power Measurement
	524	OpenSSH Multiple Vulnerabilities	Rockwell Automation, Siemens, Schneider Electric, Honeywell, AKCP, RuggedCom
	358	Terminal Services Encryption is Medium or Low	Rockwell Automation, Schneider Electric, Siemens, RuggedCom, Scalence

In addition to the high impact and exploitability of some of the vulnerabilities as the 'Critical' risk rating, Nessus was also able to identify various vulnerabilities at the 'High' and 'Medium' risk levels which provide attackers with viable attack vectors for exploiting SCADA systems. The main vulnerability at the 'High' risk level are based on the OpenSSH and DropBear SSH services. Secure Socket Shell (SSH) is a protocol designed to allow individuals to remotely access systems in a secure manner. OpenSSH and DropBear are specific SSH clients installed on systems to provide SSH capabilities to that system. Nessus identified that devices from major vendors such as Rockwell Automation, Siemens, Schneider Electric, and Honeywell provide PLC's, HMI's and building network adapters all running older versions of OpenSSH and DropBear. These older versions allow attackers to potentially exploit the SCADA systems by hijacking the

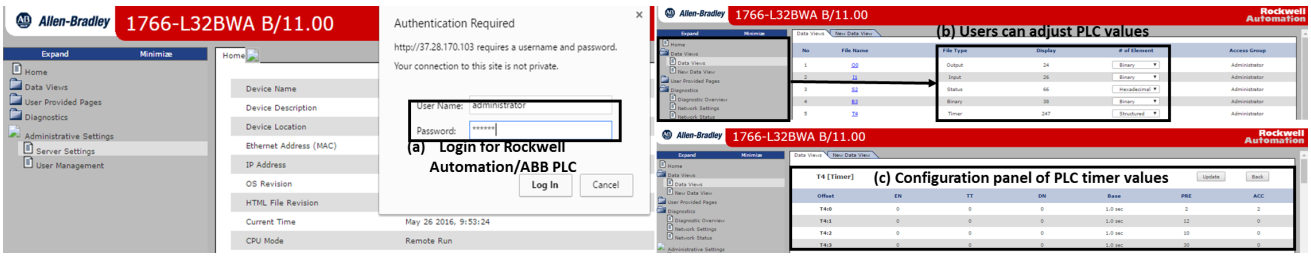

Fig. 6. Rockwell Automation PLC: (a) Logging into the PLC, (b) potential adjustment of a variety of PLC values, (c) configuration panel of timer values

SSH client and gaining remote access over the system. Nessus was also able to identify 29 Schneider Electric database and FTP SCADA systems that were vulnerable to Default Credentials.

In addition to SSH and Default Credential vulnerabilities, Nessus discovered 1,407 devices which were running unencrypted Telnet servers. These vulnerabilities were categorized at the 'Medium' risk level. Like SSH, Telnet allows users to remotely log into the system. Having an unencrypted Telnet server running on a system allows attackers to potentially conduct man-in-the-middle attacks, where they are able to alter or relay false information between the SCADA system and the owner of the device operating the device or steal device credentials. In the context of SCADA systems, this means that an attacker can send false signals from the SCADA systems to the owner, thus causing the owner to send signals which may impact what the SCADA system is controlling.

The 'Medium' risk level also contained a variety of devices susceptible to Modbus Coil Access. As previously mentioned, Modbus is a SCADA specific protocol designed to transmit information over serial lines between SCADA specific components [1]. Modbus is often used in a master-slave architecture, where SCADA components such as Master Terminal Units (MTU's) gather information from slave units (i.e., Remote Terminal Units). The Modbus Coil Access vulnerability indicates that attackers can read and adjust transmitted values. These adjustments could cause the MTU's or RTU's to send improper signals to each other, thus damaging the underlying infrastructure.

Overall, results of the active assessment indicate that a variety of devices are afflicted with serious vulnerabilities. One vendor which shows consistent vulnerabilities across all risk levels is Rockwell-Automation. However, the core reason why Rockwell-Automation is more vulnerable is unclear. Nevertheless, many of the aforementioned vulnerabilities can be readily remedied by changing the default passwords or by consistently updating the system software.

V. Conclusion and Future Directions

Although the Internet has provided many modern conveniences to the management of SCADA systems and critical infrastructure, it has also opened up these systems for potential cyberattacks. This study contributed to SCADA security literature by systematically identifying SCADA vulnerabilities on the Internet of Things by utilizing state-of-the-art passive and active vulnerability assessment techniques. Overall, we were able to identify a multitude of vulnerabilities including the use of default usernames and passwords as well as outdated software on SCADA components such as PLC's and HMI's. Such vulnerabilities afflict major SCADA vendors such as Rockwell-Automation, Siemens, and Schneider Electric.

There are many promising directions for future expansion. First, additional work can be done to better identify SCADA device owners. Second, additional SCADA devices can be identified from the Shodan database beyond what is provided from the API by using text and data mining techniques on Shodan's device banner data. Finally, longitudinal analysis can be conducted to understand how SCADA vulnerabilities change and evolve over time. All of these extensions would help to provide a deeper understanding SCADA vulnerabilities on the Internet of Things.

Acknowledgments

This material is based upon work supported in part by the National Science Foundation (DUE-1303362 and SES-1314631).

References

[1] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke, "SCADA security in the light of Cyber-Warfare," Comput. Secur., vol. 31, no. 4, pp. 418–436, Jun. 2012.

[2] M. Patton, E. Gross, R. Chinn, S. Forbis, L. Walker, and H. Chen, "Uninvited Connections: A Study of Vulnerable Devices on the Internet of Things (IoT)," in 2014 IEEE Joint Intelligence and Security Informatics Conference, 2014, pp. 232–235.

[3] White House, "Foreign Policy - Cybersecurity," 2008. [Online]. Available:https://www.whitehouse.gov/issues/foreign-policy/cybersecurity. [Accessed: 25-Nov-2015].

[4] B. Miller and D. Rowe, "A survey of SCADA and critical infrastructure incidents," in Proceedings of the 1st Annual Conference on Research in Information Technology, 2012.

[5] R. Robles, M. Choi, "Assessment of the Vulnerabilities of SCADA, Control Systems and Critical Infrastructure Systems," International Journal of Grid and Distributed Computing, vol 2, no. 2, pp. 27-34.

[6] E. Ayaburi and L. Sobrevinas, "Securing Supervisory Control and Data Acquisition Systems: Factors and Research Direction," in Americas' Conference on Information Systems (AMCIS), 2015.

[7] I. Onyeji, M. Bazilian, and C. Bronk, "Cyber Security and Critical Energy Infrastructure," Electr. J., vol. 27, no. 2, pp. 52–60, Mar. 2014.

[8] F. Daryabar, A. Dehghantanha, N. I. Udzir, Nor Fazlida binti Mohd Sani, and S. bin Shamsuddin, "Towards secure model for SCADA systems," in Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012, pp. 60–64.

[9] D. Kennedy, J. O'Gorman, D. Kearns, and M. Aharoni, Metasploit: The Penetration Tester's Guide, 1st edition. No Starch Press, 2011.

[10] A. Nappa, R. Johnson, L. Bilge, J. Caballero, and T. Dumitras, "The Attack of the Clones: A Study of the Impact of Shared Code on Vulnerability Patching," in 2015 IEEE Symposium on Security and Privacy, 2015, pp. 692–708.

[11] E. Byres, "Surviving the ICS Vulnerability Avalanche - New Technologies for New Security Threats," 2013. [Online]. Available:https://files.sans.org/summit/scada14/PDFs/Surviving the ICS Vulnerability Avalanche - Eric Byres, Belden.pdf. [Accessed: 20-Nov-2015].

[12] B. Genge and C. Enăchescu, "ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services," Secur. Commun. Networks, Apr. 2015.

[13] Rockwell Automation, "Installation Instructions - MicroLogix 1400 Programmable Controllers," 2015. [Online]. Available:http://literature.rockwellautomation.com/idc/groups/literature/documents/in/1766-in001_-en-p.pdf. [Accessed: 20-Nov-2015].

[14] SANS, "The Conficker Worm," 2008. [Online]. Available: https://www.sans.org/security-resources/malwarefaq/conficker-worm.php [Accessed 26-May-2016].