Cyberattacks on SCADA Systems

Guillermo Francia, III, David Thornton, and Thomas Brookshire, *Jacksonville State University*

Abstract--Critical infrastructures such as the Supervisory Control and Data Acquisition (SCADA) systems have succumbed to the demands of greater connectivity. Although the scheme of connecting these critical equipment and devices to cyberspace has brought us tremendous convenience, it also enabled certain unimaginable risks and vulnerabilities. These risks and vulnerabilities are very critical to our daily existence and are perilous to ignore. This paper presents an overview of the vulnerabilities of SCADA systems. Also described are proof-of-concept methods of attacking some of these vulnerabilities.

Index Terms--SCADA, Security, Wireless networks, Penetration Testing, Vulnerability Assessment, Risk.

I. Introduction

Supervisory Control and Data Acquisition systems, also known as SCADA, function at the core of our nation's critical infrastructure. A wide variety of industrial applications, including water treatment, manufacturing, and power generation all depend on these distributed industrial control systems. Each of us is affected daily by these systems at the local, regional, and national level.

Traditionally, these systems have employed wired technology for networking, when the components were networked at all. The growing prevalence of wireless technology has made supervisory control over widely distributed components a financially feasible and convenient option. However, the inclusion of wireless technology has laid bare these critical infrastructure systems to numerous cyberattack vulnerabilities.

Wireless network security has been widely studied as wired networks. However, many of the information assurance strategies advised for wired networking do not transfer readily to wireless networks. For instance, while a broadcast wireless message may be encrypted, physical security of the medium is impossible. This is exacerbated by the heterogeneity that is commonplace in wireless networks, which adds to the complexity in assuring security [1].

Guillermo A. Francia, III, David Thornton, and Thomas Brookshire: Mathematical, Computing, and Information Sciences (MCIS) Department, Jacksonville State University, 700 Pelham Rd N, Jacksonville, AL 36265 USA

From government agencies to private commercial industries, computer and information security is not only being taken more seriously but has become one of the primary foundations for the software development lifecycle itself. And yet, as this increased awareness of security minded development expands, there are still many developers who do not have the expertise or experience to properly secure their programs. This concern has led to the automation of some forms of vulnerability analysis and penetration testing in order to allow even new and learning developers to create more secure applications. One such category of these tools is known as fuzz testing.

A. General Wireless Threats

I. Shared medium

As aforementioned, physical security of the wireless medium is impossible. Similar to Ethernet, this weakness allows hackers to monitor traffic not intended for them by setting their network adapters to "promiscuous" mode. Not only does this leave the nodes of such a network vulnerable to loss of confidentiality, but also to "man-in-the-middle" attacks, wherein a hacker eavesdrops on messages between two or more nodes and relays or modifies messages so that the legitimate nodes are deceived into thinking they are talking directly to one another [2]. To protect from such attacks, a successful information assurance strategy must make the mere reception of a signal useless to a would-be hacker.

II. WEP cracking

While WEP (Wired Equivalent Privacy) was officially deprecated by IEEE in 2004, many legacy systems still depend on this weak security algorithm. A hacker with very little technical skill can ascertain the network key in minutes using freely available software [3], thereby gaining access to the network. To ameliorate this problem, IEEE released Wi-Fi Protected Access (WPA) as a solution. The newest version, WPA2, employs the Advanced Encryption Standard (AES), but it only works with newer access points. Thus, many legacy systems cannot take advantage of its improved security.

III. WPA and WPA2 cracking

Even though the security provided by these algorithms is much stronger than WEP, cracking them is still possible

because of the human element. For instance, the WPA specification states that a passphrase has about 2.5 bits of security per character [4]. Thus, a passphrase of less than around 20 characters would not generate a key powerful enough to deter serious attacks. Most end users opt for a shorter password that is easier to remember (often, a dictionary word). Even for stronger keys, a brute force dictionary search can be run offline to discover the plaintext key. If such searches are calculated by a large number of hackers who share their computed data (such as the publicly available list from "the Church of Wi-fi" at http://www.churchofwifi.org/), such an attack becomes feasible. The rise of cloud computing has made such attacks even stronger. Sites like wpacracker.com offer customers a wealth of processing power for a relatively cheap price. For a mere $17, customers can send their captured handshake for a search through a 135 million word dictionary optimized specifically for WPA passwords. In only 20 minutes, the site can complete a search that would take an average dual-core laptop 5 days.

IV. Resource-depletion and Denial of Service Attacks

Because wireless networks often suffer from much lower bandwidth than their wired counterparts, system availability can be harder to maintain even without cyberattacks. Wireless signals are especially vulnerable to jamming. For instance, a global positioning system (GPS) jammer can be constructed for around $30 with equipment obtainable from most electronic supply stores. In fact, such a jammer can overpower legitimate signals in up to a 75-mile radius [5]. Another means of blocking service is resource depletion. By submitting multiple phony authentication requests to an access point, an attacker can overwhelm its resources [6], preventing legitimate clients from using the access point. Such an attack can also facilitate the introduction of impostor access points, as discussed in the next section.

V. Rogue access points

Many wireless cards have the ability to operate as an access point. As the Service Set Identifier (SSID) of a network can be arbitrarily set, it is easy for a hacker to impersonate a legitimate Access Point (AP) by simply copying its name. Because 802.11 authentication is one-way (from AP to client), clients could attempt to connect to this rogue access point [7]. Most clients are set up to continue connecting to a network they have previously logged into, especially if it has the strongest signal available. Further, if the hacker knows how to login to the legitimate AP (often, because it is not secured), he can commit the aforementioned "man-in-the-middle" attacks. This way, the hacker can covertly monitor information between the nodes (threatening confidentiality), modify the information before passing it along (threatening integrity), and (in the case of SCADA systems) forward bogus commands which may put the Remote Terminal Units (RTUs) into an unusable state (threatening availability). Impersonating an open access point is easier still with free, publicly available software like Karma, which can simultaneously spoof several legitimate APs [8].

VI. Injection attacks

Because wireless is a shared medium, hackers can view legitimate traffic with a freely available packet sniffer. If the access point is open, it is easy to quickly read and reply to the message with a fake reply. With freely available packet injection software like Airpwn, an unscrupulous user can send modified versions of legitimate requests before the authentic web server has a chance to respond. When the legitimate reply arrives a moment later, it is rejected by the client as erroneous [9].

VII. Analog wireless eavesdropping

Many wireless devices are neither digital nor encrypted. Headsets, wireless phones, and wireless microphones are often transmitted "in the clear", which makes eavesdropping easy with inexpensive scanners such as KeyKeriki [10]. Even if the attacker is only intercepting daily office conversation, victims are providing the attacker a foothold for later social engineering techniques. Note that even digital versions of these devices may not be encrypted, and are thus only marginally more secure. Encrypted versions, however, can cost several times more than their simpler legacy counterparts. Video eavesdropping does not fare much better, as most surveillance cameras broadcast without authentication. If an attacker is willing to invest some time, he/she can collect a wealth of data that can be later leveraged in more destructive and invasive attacks.

The next section discusses a variety of SCADA security threats.

II. SCADA Security Threats

Unfortunately, many industry professionals view security as an afterthought. Security measures such as firewalls, encryption, and logging are often disabled when troubleshooting a system; often, these measures are not re-enabled, either due to convenience or forgetfulness. Even if security measures are active, administrators often neglect to update software and/or firmware regularly.

A. Standardization

The move from proprietary to standardized software has made systems more vulnerable, since the weaknesses of the operating systems are well-documented. Thus, older SCADA systems benefited somewhat from "security through obscurity", because an attacker would have to be educated about that particular vendor's software and hardware eccentricities [11]. Recently, the rise of standardized protocols has ironically allowed attackers to focus more intensely on a smaller set of better documented protocols. One of the most popular protocols is the freely available Modbus, which is used on millions of devices. Unfortunately, its design goals of interoperability and convenience also make it inherently vulnerable to attack [12]. Some of the other popular protocols in use today include DNP3, EtherNET/IP, PROFIBUS and Foundation Fieldbus. Many of these designers create their SCADA protocols with a closed system in mind, which is rapidly changing [13].

B. Shared networking

In order to benefit from shared resources, many city organizations operate on the same network. One or more of the network nodes likely connects to the Internet at some point. Hackers who have discovered a way in through one of these nodes can leverage their way in to the SCADA system as well.

C. Low-speed connections with long polling cycles

Because many changes in a SCADA system happen slowly, high-speed communications are often unnecessary. Thus, many such systems employ slower, less expensive technology such as analog radio, with polling cycles as long as several minutes. This means that the system only communicates with each RTU a handful of times per hour. A hacker can take advantage of this by attacking the RTU immediately after it has finished communication, confident that he can operate with impunity while the base station polls other RTUs [14].

D. Alternative communications technology

To maximize reliability, most industrial systems provide redundant communications technologies. In the event that one type fails, a secondary system can function while the first is restored to working order. Often this takes the form of a phone of a dial-up telephone line, which is vulnerable to a host of security attacks. Therefore, a clever hacker could perform a double-pronged attack: a denial-of-service (DoS) attack on the primary technology (such as the radio system), then a secondary attack on the weakly secured secondary technology.

E. Radio communication

In radio-based communication, many SCADA systems feature an omni-directional base station antenna, while the individual nodes employ directional (yagi) antennas. This means that an attacker who is in the system's radius can easily intercept the base station's messages, but may not hear the nodes. With some time and persistence, an attacker can simply drive around town until he/she can detect the nodes as well. Once both sides of the communication can be heard, it is not difficult to determine the protocol and the system setup. At this point, the attacker is poised to perform man-in-the-middle attacks, with neither the node nor the base station being aware of the intrusion.

D. Easily Programmable RTUs

Once the attacker has achieved the aforementioned condition, his/her job is once again made easier by SCADA features. In order to make system adjustments without having to visit each node physically, many SCADA systems allow on-the-fly programming of RTUs through the same communications channel used for routine message passing. This vulnerability allows an attacker to download, evaluate, and reprogram the RTU. At this point, an attacker can shut off or burn out motors, fake sensor data, etc. A clever programmer can even insert their own insidious code into what appears to be a normally operating system.

III. Proof of Concepts

We describe some of the vulnerabilities found in wireless SCADA systems and demonstrate successful attacks through proof-of-concepts. We applied these proof-of-concepts on an existing SCADA testbed known as the Critical Infrastructure Security and Assessment Laboratory (CISAL). The laboratory, which is fully described by Francia, et al. [15], is designed to emulate, as well as simulate, the control systems that are prevalent in real-world critical infrastructures. The installed equipment, found in the laboratory, includes a variety of hardware (both legacy and state-of-the-art) from various manufacturers. The activities and projects in this laboratory are designed and structured to provide practical experiences while illustrating theory in the pertinent research areas.

A. Cellular Communications (Modems)

Modern cellular networks are currently being adopted at an unprecedented pace by utility companies to provide the communication functions needed to manage vital control systems. Cellular system providers now offer data services over the 3G networks such as Wideband Code

Division Multiple Access (W-CDMA) and High Speed Downlink Packet Access (HSDPA). With the introduction of 4G technologies to work with the newer mobile devices, Long-Term Evolution (LTE) technologies is projected to have an impact on the remote management of control systems as well.

Although these publicly accessible cellular modems are provided with built-in security features such as IPSec-compliant Virtual Private Network (VPN) and firewall capabilities, vulnerabilities still exist. The CISAL testbed is configured with an active cellular connection to control a simulated waste treatment plant. We describe a number of proof-of-concepts that can be used to attack these vulnerabilities in the testbed.

The default administrative username and password for the cellular modem are "user" and "12345", respectively. Most cellular users do not want a very complicated password and thus, would opt to something that can easily remember. We experimented on using several dictionary words as passwords and with a widely available password cracking software such as Hydra, the administrative privilege is easily compromised. Figure 1 depicts the user interface for the cellular modem configuration and could easily be reached by an intruder because of a weak password. It should be noted that this interface provides complete control of the functionalities and security features of the cellular modem.

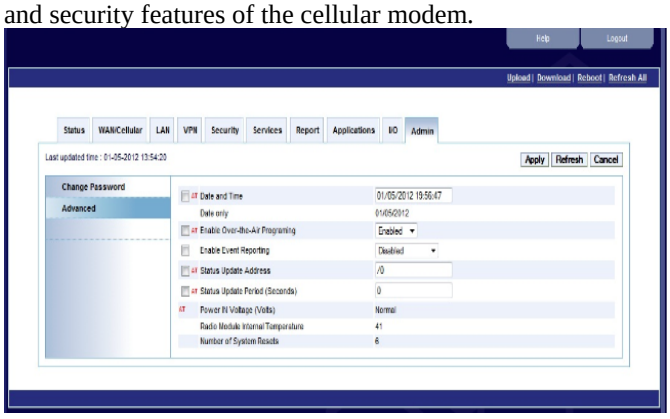

Figure 1. Cellular Modem Configuration

B. Industrial Radios (900 MHz)

High speed frequency hopping Ethernet data radio communications on the license free 900 MHz and 2.4GHz bands have also been deployed for telemetry and SCADA applications in the water, wastewater, gas, oil, and mining industries. Most of these Ethernet radios provide a point-to-point or point-to-multipoint setups and can transmit at a speed of 512 kbps at a distance of at most 45 miles.

The J-Series TRIO Radio [16], manufactured by Schneider Electric, employs several layers of security. The first layer uses the network name for deriving the hopping sequence. The second layer of protection uses the serial numbers as trusted labels. The third layer is through the use of a 256-bit AES encryption key to encrypt the data that is to be transmitted.

The vulnerability of this system is found in the authentication interface. A proof-of-concept attack is to first use nmap to scan the default network. After discovering the IP address of a TRIO radio modem, the Hydra password cracker is used to crack the administrative password. With a weak password, Hydra can easily complete the discovery process and thus, provide full access to the TRIO radio configuration.

C. Wi-Fi Networks (802.11)

Vulnerabilities of Wi-Fi (802.11) networks are well documented. Descriptions of well known vulnerabilities and corresponding attacks are found in [9] and [17]. Wi-Fi networks are prone to Man-In-The-Middle (MITM) attacks. Using the CISAL testbed, we carried out a passive MITM attack and a Denial-of-Service (DOS) attack.

In the passive MITM attack scenario, Host A (Remote Controller) is communicating with Host B (Programmable Logic Controller (PLC)) through an 802.11 router. Host C (Attacker) has positioned itself as the MITM attacker through ARP poisoning. ARP poisoning is carried out to force changes in the ARP cache entries in the two communicating hosts to trick them that they are directly talking to each other. The fact is, through the poisoned cache entries, their communications are being routed through a middle host, the MITM attacker. In order to properly take a message and forward it to the other party, the attacker must enable IP Routing through the registry settings. A freely available tool, Cain and Abel [18], is used to gather the ARP information on the network and then successfully poison the victims ARP cache entries. The conversation between Hosts A and B can then be viewed in Wireshark as redirected packets. The proof of concept is carried out using a Java implementation of the User Datagram Protocol (UDP).

In the DOS attack we used Ettercap [19] for ARP poisoning and MITM. We first created a filter with the IP address of the controller and the three Ettercap functions:

drop(), kill(), and msg(). We compiled the filter with Etterfilter and executed Ettercap. We managed to bring down the controller and totally disabled it.

IV. Conclusion and Future Plans

We have presented an overview of wireless security and vulnerabilities of SCADA systems. Using the CISAL testbed, we also have demonstrated, through proof-of-concepts, the ease of the attacking some of these vulnerabilities. The freely available tools used in these proof-of-concepts further underscores the importance of security diligence and the urgent need for vulnerability mitigations in our critical infrastructure systems. These vulnerabilities and proof-of-concept attacks clearly validates the notion that security problems found in generic computer systems occur in industrial control systems as well. However, communication protocols and technologies which are endemic to control systems are not immune to cyber-attacks.

Recognizing these needs, we plan on expanding this research project by investigating the following:

- Vulnerability assessment of advanced industrial protocols such the Common Industrial Protocol (CIP), the Object Linking and Embedding (OLE) for Process Control (OPC), the Modbus TCP, and Distributed Network Protocol (DNP3);
- The possibility of crafting packets using the Colasoft Packet Builder [6] to enable fuzz testing of controllers and switches.
- Development of security best practices for critical infrastructures; and
- Collection and analysis of forensic data from critical infrastructure networks.

V. Acknowledgements

This paper is based upon a project partly supported by the Department of Defense-National Security Agency under grant number H98230-10-1-0419 and the National Science Foundation under grant award OCI-0959687. Opinions expressed are those of the author and not necessarily of the granting agencies.

VI. References

[1] Zheng, P., Peterson, L., Davie, B., and Farrel, A. (2009) Wireless Networking Complete. Boston, MA: Morgan Kaufmann.

[2] Ahmad, A. (2005) Wireless and Mobile Data Networks. Hoboken, NJ: Wiley & Sons.

[3] Aircrack-ng Website (2011). Simple WEP Cracking with a Flowchart. Retrieved October 30, 2011, from http://www.aircrack-ng.org/.

[4] IEEE (2004). IEEE 802.11i WPA Specification. Retrieved October 29, 2011, from http://standards.ieee.org/getieee802/download/802.11i-2004.pdf.

[5] Kipper, G. (2007) Wireless Crime and Forensic Investigation. Boca Raton, FL: Auerbach Publications.

[6] Martinovic, I. et al (2008) Wireless Client Puzzles in IEEE 802.11 Networks: Security by Wireless. Proceedings of WiSec'08, pages 36-45, March 31-April 2, 2008.

[7] Chandra, P. (2005) Bulletproof Wireless Security. Boston, MA: Newnes Publishers.

[8] Karma Project (2011). Linux Rogue Access Point HOWTO. Retrieved October 22, 2011, from http://wirelessdefence.org/Contents/RogueAPHowtoBASICLinux.htm.

[9] Haines, B. (2010) Seven Deadliest Wireless Technologies Attacks. Boston, MA: Syngress Publications.

[10] Moser, M. (2007) 27Mhz Wireless Keyboard Analysis Report. Retrieved November 1, 2011, from https://www.dreamlab.net/files/articles/27_Mhz_keyboard_insecurities.pdf.

[11] Cagalaban, G. (2009) SCADA Network Insecurity: Securing Critical Infrastructures through SCADA Security Exploitation. Journal of Security Engineering, Vol. 6, No. 6, p. 473-482.

[12] Byres, E. (2005) The Use of Attack Trees in Assessing Vulnerabilities in SCADA Systems. Retrieved November 7, 2011, from http://www.ida.liu.se/~rtslab/iisw04/camready/SCADA-Attack-Trees-Final.pdf.

[13] Shaw, W. (2006) Cybersecurity for SCADA Systems. Tulsa, OK: PennWell Corporation.

[14] Francia III, G. A., Bekhouche, N., and Marbut, T., (2011). "Implementation of the Critical Infrastructure Security and Assessment Laboratory (CISAL)," Proceedings of the 2011 International Conference on Security and Management (SAM'11) . July 18-21, 2011, Las Vegas, NV. 2011.

[15] Schneider Electric, "J Series: Spread Spectrum Ethernet Radio JR900 & JR240" Retrieved January 10, 2012 from http://www.trio.com.au/p-j-over.php

[16] Cache, J., Wright, J. and Liu, V. (2010) Hacking Exposed Wireless: Wireless Security Secrets and Solutions, 2nd Edition. McGraw-Hill Companies.

[17] Sanders, Chris (2010), "Understanding Man-in-the-Middle Attack," Retrieved January 10, 2012, from http://www.windowsecurity.com/articles/Understanding-Man-in-the-Middle-Attacks-ARP-Part1.html

[18] Ettercap (2012). Website: http://ettercap.sourceforge.net. Retrieved: February 18, 2012.

[19] ColaSoft (2012). "Colasoft Packet Builder." Retrieved January 15, 2012 from http://www.colasoft.com/packet_builder.