International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.2, Issue No.8, pp : 756-758 1 Aug. 2013

Critical State Based Filtering System with Code Security for SCADA Network Protocols

B. Dineshshankar[1], N.Rajeshkumar[2], S.Thamaraichelvi[3]

[1]Department of Applied Electronics, Sasurie College of Engineering, Tirupur, Tamilnadu, India.

[2]Assistant Professor/ ECE, Sasurie College of Engineering, Tirupur, Tamilnadu, India.

[3]Department of Applied Electronics, Sasurie College of Engineering, Tirupur, Tamilnadu, India.

dinshan555@yahoo.co.in, rajeshkumar.n1@gmail.com, thamuju13@gmail.com

ABSTRACT

SCADA (supervisory control and data acquisition) is a type of industrial control system (ICS). It's a centralized system that monitors and control industrial processes that exist in the physical world. They work in Master – Slave basis. This system is widely used in Power plants, traffic light control, power plants, etc., As it is a centralized system storing lot of data, there is chance for the attackers to hack the information. In the existing system, a special filtering system is used which acts as a firewall for the SCADA network. System is prevented from hackers by analyzing the state of the system. It involves the prediction of finding whether the system is close to the critical state.The problem is that only attack by the hacker is prevented. Any problem in the software of PLC (Programmable Logic Controller) is not identified. There is no protection against unintentional errors or and other code level attacks. Some errors will show only warning, the user will ignore the warning and upload the malicious code into the server thus affecting the entire system. We focus on software vulnerabilities in ladder logic; a popular graphical language for programmable logic controllers. We show how intentional or unintentional errors in the ladder logic code can lead to integrity and availability violations. We propose methods to support secure Programmable logic controllers code development and to detect vulnerable applications.

Keywords : SCADA, ICS, PLC, filtering

I. INTRODUCTION

INTRODUCTION

While Supervisory Control and Data Acquisition (SCADA) systems have been employed to monitor and control industrial facilities for decades, the designs of these systems, their components, and the communications protocols are primarily proprietary. There has been a trend of late to define standard interfaces and communications protocols, primarily driven by the growth of the Internet and consolidation of utility companies and industries. These efforts are a means of providing cross-vendor compatibility and modularity: since replacing an existing SCADA network is an expensive proposition, integrating existing systems is the most economical approach. Further, communications between nodes in a SCADA system has been, until recently, over closed networks. With the advent of the Internet, many SCADA monitoring and control networks have been connected, at some level, to open networks, thereby inheriting all the problems and concerns associated with nodes on the Internet. Because of these two trends, there are concerns about the security and stability of SCADA systems, especially since the September 11, 2001 attacks. Recent failures of critical infrastructure SCADA systems, such as the North East blackout in August 2003, highlight these concerns. Most of our nation's infrastructure is controlled in one way or another by a SCADA system, and the Department of Defense (DoD) relies heavily on the existing commercial infrastructure for its operations. To ensure continued operations in times of crisis, the SCADA systems on which the DoD depends must be secured. An abstract generic framework for defining and understanding SCADA systems is needed as a first step toward securing them. SCADA systems are not designed with security in mind; rather the priority for developers has been reliability, availability, and speed. This does not mean they cannot be secured, however. If we can understand a particular system's features, functions and capabilities, we can address its limitations. A generic abstract framework provides a tool to understand the system's features, functions and capabilities, and how components in the system relate and interface with each other. With that information about the system, we can begin the process of securing it. It describes the different components in a SCADA system and the variety of open communications protocols that have been defined. The thesis then refines a three-tiered model and ultimately provides a matrix approach to describing and defining the features, functions and capabilities of a SCADA system. To address vulnerabilities present in the power system, NERC, working with the DOE and the DHS and their Canadian counterparts, has developed a set of cyber security standards. These standards are a protocol requiring companies to identify their vulnerabilities and risks and take steps to mitigate them. This is done in several steps. First, the impact of the loss of assets is determined. Next, the standard calls for the identification of vulnerabilities. Then, using this information, companies can preform risk analysis to decide which

vulnerabilities are most important to protect against. Finally, companies decide which defenses are most cost effective and begin to implement them. Common defences against cyber attacks include application of firewalls and authentication methods. he network client sends packets to the PowerWorld server via a proxy server on a specified, arbitrary port (in this case port 2001). The proxy server then translates the destination of these packets to the virtual IP address of the PowerWorld server in the simulated network. The packets are then delivered through a VPN tunnel to the RINSE node. A Network Client PowerWorld Server Port 2002 Port 2001 Proxy + VPN Client Proxy + VPN Client RINSE + VPN Server VPN Tunnel VPN Tunnel Fig. 5. Client-Server-RINSE Integration Scheme TABLE I SCADA SERVICES MAPPING Modbus Device PowerWorld Service Branch RTU open/close transmission line Generator RTU read generator information daemon grabs the packets from the RINSE end of the VPN tunnel and injects them into the simulator using the emulation capability of RINSE. RINSE then simulates a large network in which there are virtual nodes representing the PowerWorld Server and the Network Client(s). Upon arrival of the packets to the virtual node representing the PowerWorld server, the simulator generates real packets with virtual IP addresses and delivers them to the kernel. These packets are sent trough another VPN tunnel to proxy server. Finally, the proxy server translates the virtual address and sends the packets to the real PowerWorld Server. The same process happens in the reverse direction when the PowerWorld Server responds to the Network Client requests.

EXISTING SYSTEM

In the existing system they introduce SCADASim, a framework for building SCADA simulations. It provides a modular SCADA modeling tool that allows real-time communication with external devices using SCADA protocols. This work builds on the basic simulator. It adds value on top of our previous work by making SCADASim truly flexible for connecting real and simulated devices. As it is a centralized system storing lot of data, there is chance for the attackers to hack the information. In the existing system, a special filtering system is used which acts as a **firewall** for the SCADA network. System is prevented from hackers by analyzing the state of the system. It involves the prediction of finding whether the system is close to the critical state. If it is so, the filtering system will block the packets.

Disadvantages

- It is of high costs
- Security and privacy
- Only attack by the hacker is prevented. Any Software problem is not rectified.

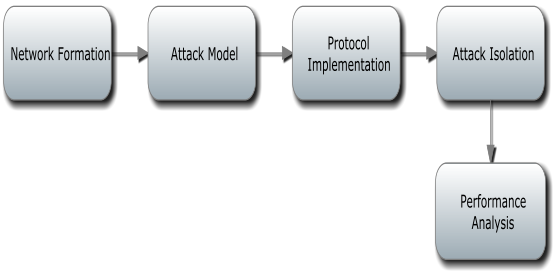
PROPOSED SYSTEM

We focus on software vulnerabilities in ladder logic; a popular graphical language for programmable logic controllers. We show how intentional or unintentional errors in the ladder logic code can lead to integrity and availability violations. We propose methods to support secure Programmable logic controllers code development and to detect vulnerable applications. Use the knowledge of the users and their intention as distinguishing characteristics with respect to the risk they represent to the SCADA system. Major Threats include: duplicate objects (i.e., objects that have been defined more than once in the PLC code), logic errors (i.e., state transition, timing, control and data flow problems), syntax errors (i.e., warnings during compilation), coding standard violations (i.e., duplicate code and tag designations, code fragmentation, etc.), unused objects (i.e., objects which were defined but were never used in the ladder logic). Minor threats include: scope and linkage errors (i.e., errors dealing with the deletion of, or failure to install, a communication block between two or more separate ladders in a PLC program), hidden jumpers (i.e., effectively bypassing a portion of a rung in a ladder logic).

Advantages

- Verifies success or failure of an attack.
- Lower entry cost.

BLOCK DIAGRAM



MODULES LIST

- Network Formation
- Attack Model
- Protocol Implementation
- Attack Isolation
- Performance Analysis

MODULES DESCRIPTION

1. NETWORK FORMATION

In this model we propose a network architecture with nodes of 300.Simulated Area is about 2km * 2Km.We Initialize the node size, position, color in the network. Vary the node speed from 5 to 30 m/s.

2. ATTACK MODEL

To check the robustness of our algorithms we need to create intruders inside our network. A node is marked as capture node and replicated all over the network. Malicious nodes are

created for disturbing the whole networks to test our environment**.**

3. PROTOCOL IMPLEMENTATION

Location Information Exchange protocol Through this a node can easily detect whether information exchanged by the node is wrong or not. By this can identify the clone node. Time Domain Detection & Space Domain Detection Scheme Both to detect node replication attack in our network by considering time and location information of nodes.

4.ATTACK ISOLATION

Once the attack is framed in our network we need to isolate this. Based on attack tolerance level we need to make decision for isolating the node from a network. If it is in above said threshold then need permanent isolation, below means no isolation and medium means temporary isolation needed.

5. PERFORMANCE ANALYSIS

Finally in the analysis phase we have analyzed the following,

- Packet Delivery ratio
- Packet overhead
- Routing cost

CONCLUSION

SCADA systems are not designed with security in mind; rather the priority for developers has been reliability, availability, and speed. The results of the tests conducted on a prototype implementing the described approach demonstrated the feasibility and validity of the proposed method. We focus on software vulnerabilities in ladder logic; a popular graphical language for programmable logic controllers. We show how intentional or unintentional errors in the ladder logic code can lead to integrity and availability violations. We propose methods to support secure Programmable logic controllers code development and to detect vulnerable applications.

REFERENCES

i. T. G. Lewis, Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation. Hoboken, NJ: Wiley, 2006.

ii. R. Krutz, Securing SCADA systems. Indianapolis, IN: Wiley, 2006.

iii. M. Brundle and M. Naedele, "Security for process control systems: An overview," IEEE Security Privacy, vol. 6, no. 6, pp. 24–29, Nov. 2008.

iv. M. Masera, I. Fovino, and R. Leszczyna, "Security assessment of a turbo-gas power plant," in Critical Infrastructure Protection II, ser. IFIP Advances in Information and Commmunication Technology. New York: Springer, Jan. 2008, vol. 290, pp. 31–40.

v. K. Munro, "SCADA—a critical situation," Network Security, vol. 2008, no. 1, pp. 4–6, Jan. 2008.

vi. J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in Critical Infrastructure Protection, ser. IFIP International Federation for Information Processing, E. Goetz and S. Shenoi, Eds. New York: Springer, 2007, vol. 253, pp. 73–82.

vii. K. Poulsen, "Slammer worm crashed Ohio nuke plant network," 2009 [Online].

viii. N. Falliere, L. O. Murchu, and E. Chien, W32.Stuxnet Dossier, Symantec Tech. Rep. 1.4, 2011.

AUTHOR'S PROFILE

First author- B.Dineshshankar. He received his B.E. degree in Electronics and Communication engineering from reputed college of Anna University, India. He is pursuing his M.E in Applied Electronics from Sasurie College of Engineering-Affiliated to Anna University, Tamil Nadu, INDIA.

Second author- N.Rajesh Kumar. He received his B.E. degree in Electronics and Communication engineering from the reputed college of Anna University, India. He received his M.E in Computer and Communication Engineering from the reputed college of Anna University, Tamil Nadu, INDIA. He is doing his research in optimized security challenges in scada systems environment. Presently he is working as Assistant Professor in sasurie college of engineering, Tirupur.

Third author- S.Thamaraichelvi. She received her B.E. degree in Electrical and Electronics engineering from the reputed college of Anna University, India in 2011. She is pursuing her M.E in Applied Electronics from Sasurie College of Engineering-Affiliated to Anna University, Tamil Nadu, INDIA. Her research interest in Network Security.