CYBER SECURITY USING MULTI-THREADED ARCHITECTURE DATA DIODE AT THE NBSR

Scott Arneson and Dağistan Şahin
National Institute of Standards and Technology Center for Neutron Research
100 Bureau Dr., Gaithersburg, MD, 20899 USA
scott.arneson@nist.gov; dagistan.sahin@nist.gov

ABSTRACT

Ongoing upgrades for the control room of the National Bureau of Standards Reactor (NBSR) located at the National Institute of Standards and Technology (NIST) require the ability to collect sensor data from the secured reactor network for trend analysis, while maintaining the security of the reactor network. To accomplish this, a data diode was created to provide a physical barrier that prevents any outside data from reaching the secured reactor network. The hardware of the data diode consists of two embedded computers, running Ubuntu OS, connected by three DB-9 serial cables. The serial cables have been modified so that only two wires, a ground and a transmission line, provide connection between the computers. The data diode relies on a multi-threaded architecture to collect and simultaneously transfer data over three serial connections. A software program, implemented in Java, has been written to request data from the Yokogawa paperless recorders installed to monitor the NBSR process instrumentation and transfer that data across the serial connections. The data will then be stored in a relational database for later analysis. The source codes for the software are available open source on GitHub.

Key Words: Data Diode, Cyber Security, NIST, NBSR, Data Acquisition

1 BACKGROUND/MOTIVATION

The NIST Center for Neutron Research (NCNR) is currently upgrading its reactor control panel and introducing modern digital components [2]. One of the goals of this upgrade is to take advantage of the new data acquisition abilities of the digital components to compile historical data in a database for further trend analysis. This requires communicating data from the secure internal reactor network with an outside network on which the database server resides. The main concern for doing this was the possibility of creating a vulnerability in the reactor network and introducing the possibility of cyber-attackers being able to access critical reactor safety components.

A solution to the cyber security problem was to design and implement a data diode. Data diodes allow communication in only one direction, much like an electrical diode allows current flow in only one direction [1]. The design goals of the data diode were to be rugged, using embedded computers with a Linux based operating system, and using custom software. The software would need to be able to communicate and gather data from the reactor data acquisition devices using Modbus communication protocol, transfer the data across the data diode, and finally store the data in the database.

1.1 Design Criteria

Though the device will be in a location that has a relatively controlled climate, the computers must be able to withstand an industrial environment. They must be able to handle increased temperature and humidity in the event of a casualty in the plant.

The data diode must be designed in such a way that data can only go in one direction. It must be physically impossible for data to travel through the data diode in the reverse direction.

It must be able to collect all the reactor data points once every second. This amounts to collecting about 1200 data points from ten different devices each second. In addition, it must be easily configurable allowing technicians to easily add or remove data points.

The software must use Modbus TCP protocol to communicate specifically with Yokogawa® paperless recorders that are currently installed in the reactor control console. Also, the software should be able to communicate generally with other Modbus devices as needed for temporary or permanent data acquisition.

The software should take advantage of multi-threading. In order to perform the data gathering and transferring, the device must perform these operations in parallel. Otherwise, it is unlikely to be able to gather all the data points every second.

2 ARCHITECTURE

2.1 Hardware

The hardware for the data diode consists of two embedded computers connected by three custom modified serial cables. The embedded computers chosen are fanless with 2.24 GHz processors, 2GB DDR3L memory, and a 500GB hard drive. They are designed for industrial applications with din-rail mount capability and a wide operating temperature range of -20°C to 60°C. The operating system that was chosen for the computers was Ubuntu 16.04 LTS. This operating system was chosen due to reliability as well as the availability of technical support.

Fig. 1 shows the high-level network architecture of the system. As can be seen, the data diode provides a path for communication between the secure reactor network to the less secure research network on which the database is stored. The "anode" side of the diode is able to communicate with the devices on the reactor network, mostly the digital recorders, via Modbus through a Modbus read-only firewall. The data can then be sent through the custom made serial cables to the "cathode" side of the diode. The cathode can then send the data to the database server on the research network, which isolates computers from the general NIST network by employing strict data and communication rules.

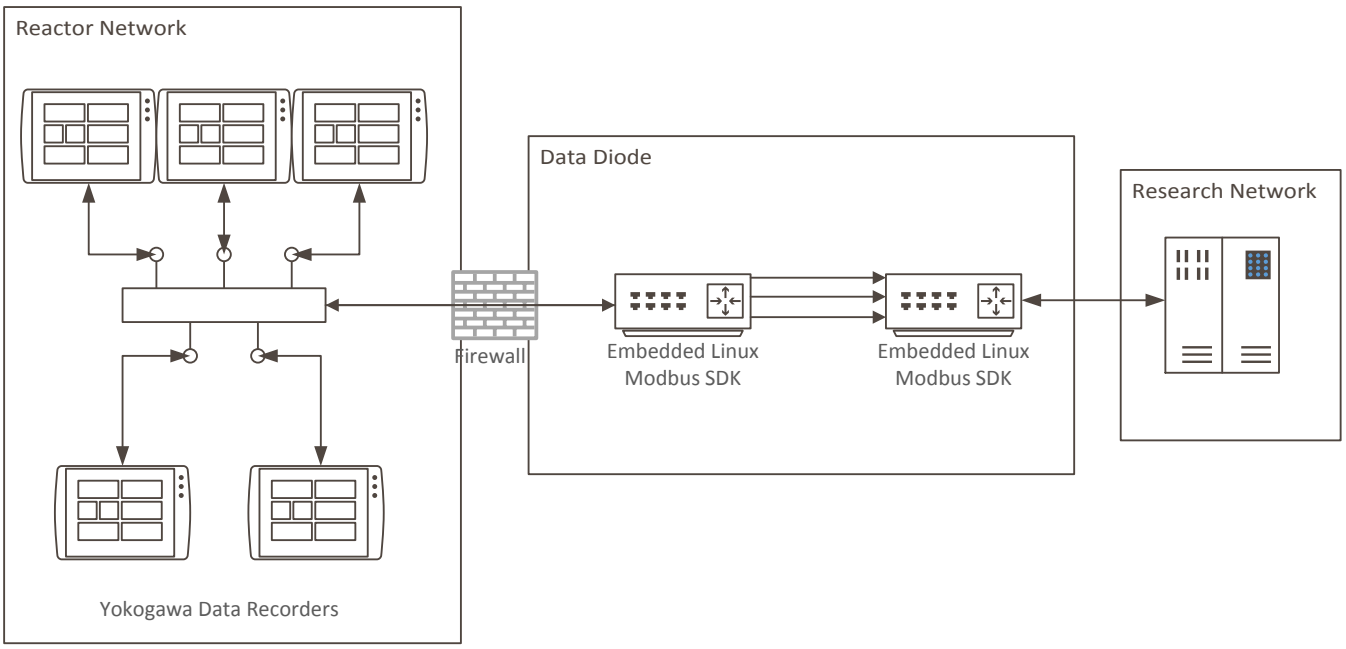

Figure 1. High level network architecture.

The communication between the computers is through modified DB9 RS232 serial cables. Normal DB9 RS232 serial cables require a minimum of three wires connecting the two ends for serial communication, a ground wire and two communication wires from each transmit pin to the opposite receiving pin. The cables used for the data diode only have two wires connecting the computers, a ground wire and one communication wire going from the transmit pin of the anode to the receiving pin of the cathode. Fig. 2 shows an electrical schematic of how the cables are connected.

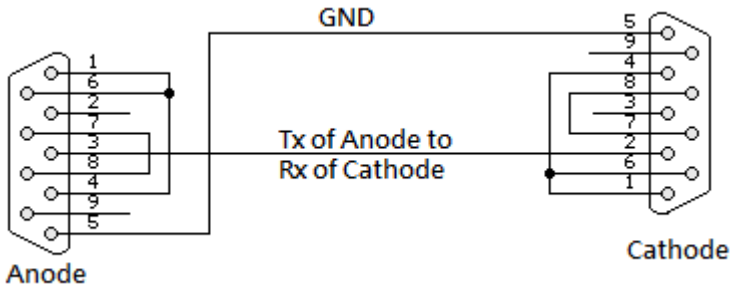

Figure 2. Custom DB9 RS-232 serial cable electrical diagram.

By only connecting the transmit pin of the anode to the receiving pin of the cathode, data can only flow from the anode to the cathode. Data flowing in the opposite direction is not possible due to no path from the transmit pin of the cathode to the receiving pin of the anode. In the unlikely scenario that the cathode was to be compromised by a malicious programmer and its receiving pin was reconfigured to be a transmit pin, then data would still be unable to flow from the cathode to the anode. Without physical access to the anode, there would be no method for an outside influence to make the transmit pin of the anode into a receiving pin. Therefore, the data diode provides a physical barrier that prevents access to the Reactor network [3].

2.2 Software

The data diode software that runs on the embedded computers is a custom Java program. The program is designed to handle the gathering of data from the digital recorders and other Modbus enabled devices, transfer data across the diode, and then store the data on a relational database. The program uses multithreading to increase speed and is designed to be easily run and configured by technicians using configuration files and the Graphical User Interface (GUI).

The anode runs five threads simultaneously. One thread begins the program, reads the configuration files, and operates the GUI. Another thread takes care of the gathering of the data from the digital recorders and other Modbus devices. Once a second it gathers data from all the devices and places the data in a globally shared First-In-First-Out (FIFO) buffer. Three other threads, one for each serial cable, wait for data to be available in the FIFO buffer and then serialize the data and send it across the serial connection.

The cathode module also runs five threads simultaneously. As with the anode, one thread begins the program, reads the configuration files, and operates the GUI. Three threads wait for data to arrive on the serial connections. When data arrives, these threads de-serialize the data and place the data on globally shared FIFO buffers. The last thread will gather the data from these buffers and send the data to be stored on the database. See Fig. 3 for a flow diagram of the program operation.

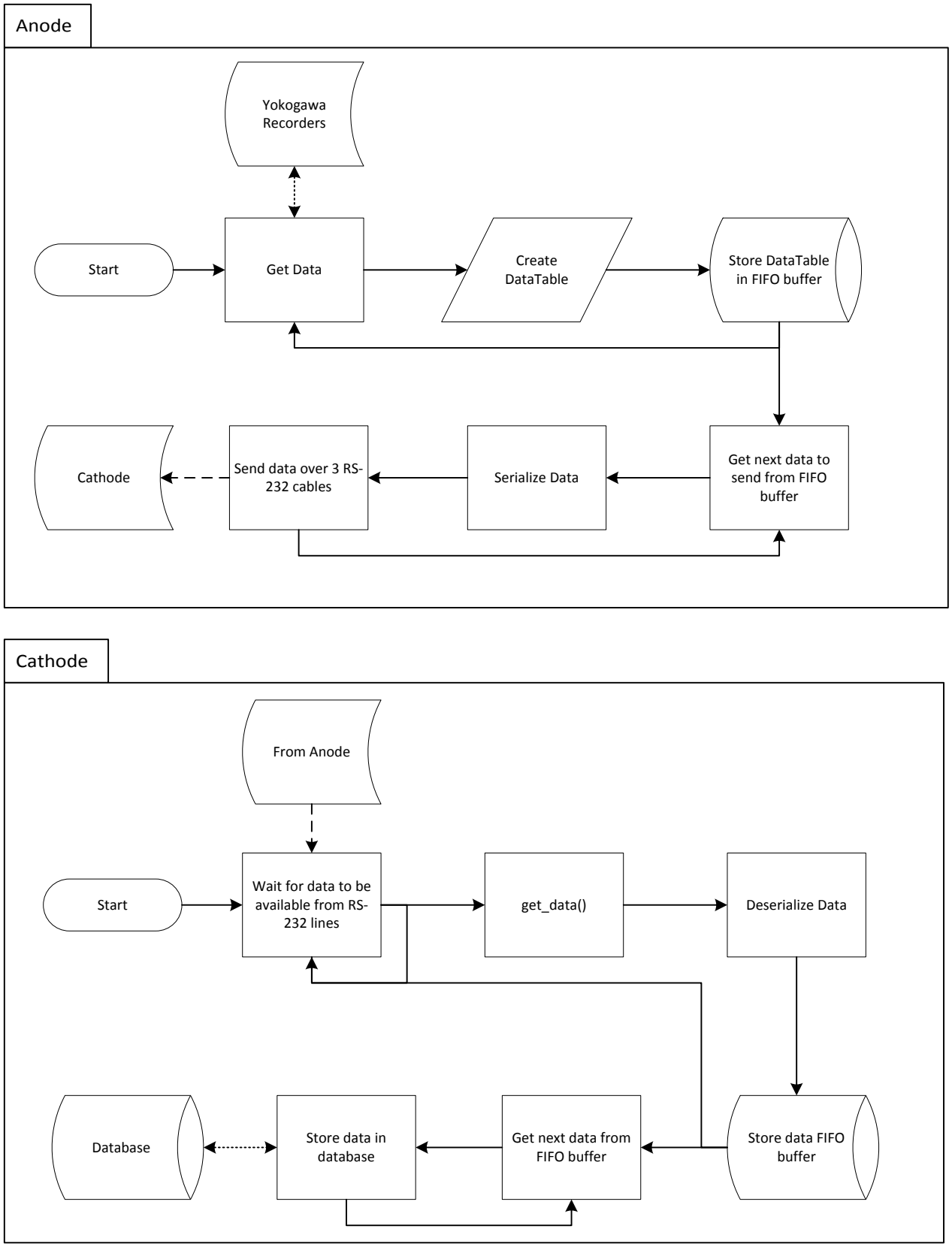

Figure 3. Flow diagram of the software running on the data diode.

The program is designed to be easily configurable. There is a main configuration file for the program that allows the technicians to list all the devices from which the program needs to gather data. For the digital recorders, which are the main devices that are to be read, the technician needs only to list a name for the recorder, the model, the IP address, and the location of the recorder's configuration file. During startup of the program, the program will create objects for each digital recorder and parse through the configuration file to determine exactly which data points need to be gathered from that recorder. For other Modbus devices, a separate configuration file was developed that allowed technicians to specify each device, its IP address, and the data points that must be gathered using their register reference numbers. This configuration file was designed to be very general and is able to allow the software to communicate with any Modbus TCP enabled device.

The program is designed such that it will automatically create and maintain the tables within the relational database. Upon connecting to the database, the program ensures that the necessary tables exist within the database and will create new ones if needed. Even if the database is completely empty at the start of the program, all the necessary tables will be created. This ensures that the program never encounters a problem due to operator error in the creation of the database. It also allows technicians to easily add new devices or data points to the system without having to worry about changing the database. The software automatically creates and adds to the necessary database tables. To add new data points, technician will only have to update the appropriate configuration files on the embedded computers and restart the program. Fig. 4 shows the architecture of the relational database and indicates important components such as primary and foreign key connections.

In the creation of this software, two different application program interfaces (API's) were used. The first is Javolution [www.javolution.org]. Javolution offers several Java classes that allowed the data to be gathered efficiently. Javolution also takes care of many of the problems when passing data in multithreaded systems by using concurrent algorithms which makes writing the code much easier. The other API that was used in the making of this software was the FieldTalk Modbus Master Java Package [www.modbusdriver.com]. This API handled the Modbus communication protocol when communicating with Modbus devices.

The software for this data diode is available open source at https://github.com/usnistgov/DataDiode.

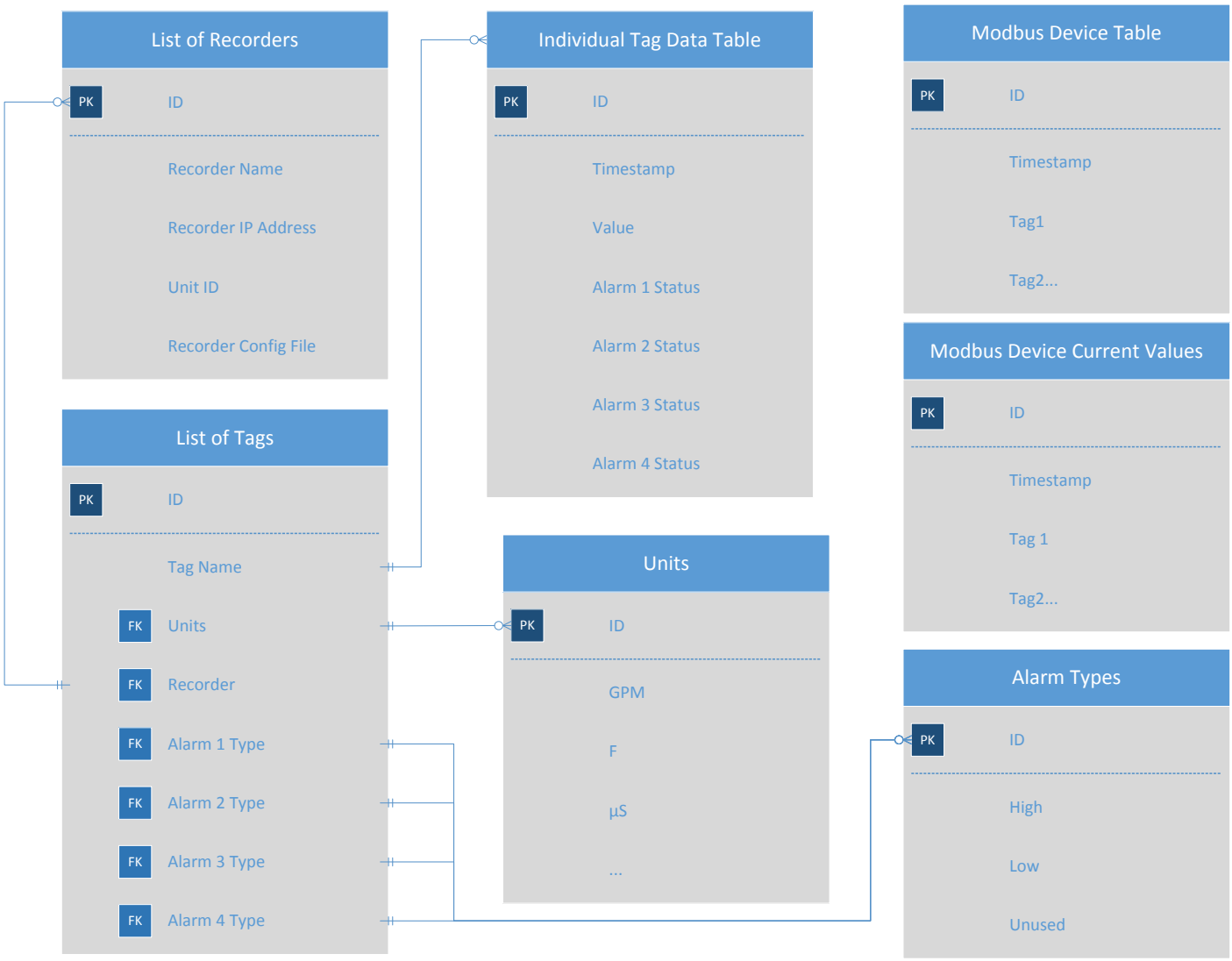

Figure 4. Relational database architecture.

3 RESULT

The data diode described above was created and successfully employed at the NCNR. It was successfully able to communicate and gather all data from the digital recorders once a second. The diode has also created and maintained a relational database, storing the data that it has gathered. The next step that is desired for the control room upgrade is to analyze the data and use a machine learning algorithm for automated trend analysis.

4 DISCLAIMER

Certain commercial equipment, instruments, or materials are identified in this study in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

5 REFERENCES

1. B. S. Jeon and J. C. Na, "A study of cyber security policy in industrial control system using data diodes," *2016 18th International Conference on Advanced Communication Technology (ICACT),* Pyeongchang, pp. 314-317 (2016).
2. D. Şahin, "NBSR Reactor Control Room Upgrade," *Trans Am Nucl Soc,* San Francisco, CA, TBP.
3. R. T. Barker and C. J. Cheese, "The application of data diodes for securely connecting nuclear power plant safety systems to the corporate IT network," *7th IET International Conference on System Safety, incorporating the Cyber Security Conference 2012,* Edinburgh, pp. 1-6 (2012)