Proc. of the 29th International Conference on Computer Application in Industry and Engineering, pp. 133 - 138, Denver, CO. September 26-28, 2016.

A Cyber-Robust Connected Control System: Experimental Validation

Charles Kim
Howard University
Washington, DC
ckim@howard.edu

Ravindranath Jaglal
Washington Square Technologies
New York, NY
ravi.jaglal@gmail.com

Abstract

This paper deals with exploitable cyber vulnerabilities in connected control systems. Even with numerous metrics and methods for intrusion detection and mitigation strategy, a complete detection and deterrence of cyber-attacks has not been found and would not be found anytime soon. Considering the impact and consequence of possible malfunctions caused by such attacks in the connected computer control systems applied to safety-critical applications, this paper proposes a new control system to assure resiliency and fail-operate even under compromised situations. The proposed new system is realized with diversification of hardware/software and unidirectional communication in alerting cyber infiltration to upper-level management. The proposed system is tested in a lab hardware experimentation setup for validation.

1 Introduction

Cyber-attacks are ever increasing as they are expanded from simple bragging intrusion to monetary gains and exploitation to trading secret theft and to military and national security espionage. One important area in the cyber-attacks in which public are not keenly aware of is connected control systems applied for smart power grid, water treatment, petro-chemical processing, mobile and home automation systems, and connected vehicles.

The connected control systems are being adopted to take advantages of remote access, and thus they no longer stand-alone but are usually connected to their corporate network via Internet or wireless network through a firewall. They provide the benefit of economy of operation; however, this relatively open connectivity has in turn resulted in an increase in security vulnerabilities [1], unauthorized intrusions into the network, data gathering, and malicious code manipulation. A successful intrusion and access into the firmware and control functions would consequently lead to disruption of normal operations and thus a public safety threat.

Presently, the hardening of system is heavily focused on the cyber security for information systems with numerous strategies and tools for anomaly and intrusion detection, network access behavior analysis, and mitigation strategy development. For control system cybersecurity, there are several common countermeasures proposed against attack vectors [2]. However, they may block some attack vectors but are not totally attack-proof. In reality, they are backward-looking metrics and measures, and are centered on post-incident analyses with subsequent damage has already occurred. As warned by the shocking Stuxnet malware attack to an Iranian nuclear facility [3], the exploitable vulnerability of connected control systems is real and, unless cyber threats are not addressed timely, there will be serious impacts to public safety and critical infrastructure.

Considering the impact and consequence of malfunctions of the connected control systems in the safety critical applications caused by cyber incidents, this paper proposes a new control system architecture so that a connected control system becomes robust and resilient even under compromised situations. The proposed system is centered on diversification of hardware and software and unidirectional communication for alerting suspicious activities so that it insensitive to variations in inputs, processes, and outputs of cyber contents. The rationale of developing a new connected control system is the plain truth that it is impossible to predict cyber events throughout the control system's lifecycle, and that detection and mitigations strategies may be good for old and known malwares and viruses only [4].

The paper is organized as follows. In the next section, we discuss about a generic but simplified connected control system and its exploitable vulnerabilities. Then, we detail the proposed architecture for new connected control systems with hardware and software diversity. In Section 4, the proposed control system is examined and validated in lab experimentation. And Section 5 concludes the paper.

2 Vulnerabilities of Existing Systems

As more components in the control systems include wireless gateways of open or standardized protocol, connected control systems invites vulnerability to the network from cyber-attacks. To perform such an attack is relatively inexpensive and the ability required to do so is not rare. Programs capable of instantiating denial of service attacks are no secret and actually available to download for free from any web-based source code repository. In addition, two way data transmissions utilized by the connected control systems would possibly allow a

978-1-943436-04-0 / copyright ISCA, CAINE 2016
September 26-28, 2016, Denver, Colorado, USA

person who understands those protocols unauthorized entry into the systems via a method known as 'Man in the Middle' attack.

To illustrate the exploitable vulnerabilities of the current connected control systems, a representative diagram for a generic control system is provided in Figure 1.

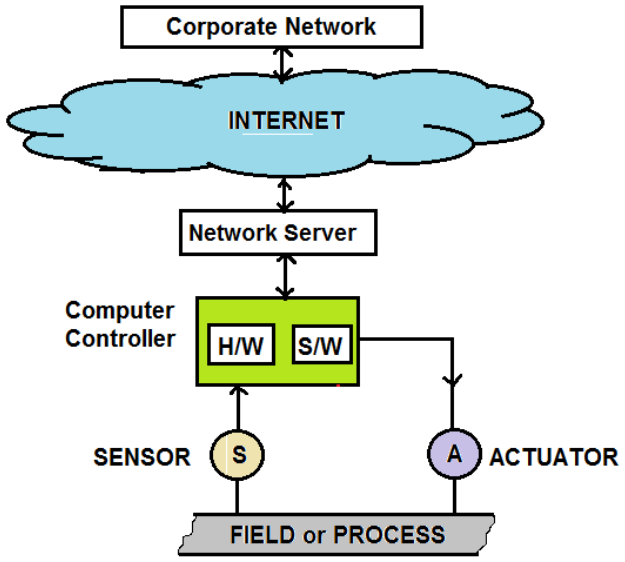

Figure 1. A typical but simplified control system.

The computer controller senses the values for the field and determines the actuator actions. The sensor may be for current magnitude or vehicle speed, and the actuator a circuit maker/breaker or vehicle's throttle or acceleration control. An enterprise-level network is connected to the control system via the Internet and communication network server. The communication network server connects other control systems in addition to the disposed computer controller.

Now consider cyber vulnerabilities of the existing control system of Figure 1. The Internet connection represents a possible entry point for hackers to infiltrate the control system. If a hacker can gather the appropriate login credentials of the communication network server, the hacker can possibly gain access to the controller and then have free reign to enact whatever change in the software code of the controller. Any alteration to the software or firmware may have major repercussions for the application the actuator is assigned to, whether it is smart-grid's power control or driverless vehicle's acceleration control.

As mentioned earlier, the present detection and prevention measures are not effective in dealing with unknown malwares and attack vectors; therefore, a new approach for cyber-robustness is required to secure connected control systems fail-safe or, further, fail-operate under compromised situations.

3 Proposed Connected Control Systems

The proposed new connected control system architecture aims to be cyber-insensitive, and the proposed control architecture is realized by diversified software and hardware and by unidirectional network connection.

3.1 Redundancy and Diversity

Here we briefly review redundancy and diversity and their differences. The use of redundancy in system design is already an accepted practice when trying to address fault and failure scenarios in software and hardware. For example, most data is typically backed up to secondary storage spaces and synchronized as often as possible to ensure minimal to no operational disturbance in most industries. Also, critical manufacturing or production processes are built with redundant hardware to allow easy replacement, repair and maintenance. However, if a computer control system is under a virus attack, for example, then even the redundant controller of the same hardware and software version as the primary controller will be susceptible to the same virus. This common-cause vulnerability would most likely ill-impact both controllers in a redundant system.

If, however, the redundant controller has different hardware specifications, there is much greater probability that the redundant controller would survive against the problem which would have caused the primary controller to fail. Added with differently designed software in it, the redundant controller with different hardware would further increase the survival and fail-operate probability. This approach of diversity, applying different hardware and software but same operational functions, seems to be the most suitable method of running a control system under the "broken part" assumption [5].

The proposed system, adopting the diversified redundant hardware and software principle, adds a supplementary controller which integrates with the existing computer controller of a connected control system, and thus makes new system fail-operate. Also, the new system adopts unidirectional communication. The structure of the proposed system is discussed next.

3.2 Diversified Architecture

A representation of the new connected control system is shown in Figure 2. In the new system, alongside the existing primary controller which is assumed to be CPU-based and thus with software codes in it, there is a secondary (duplicated) controller that functions the same in sensing and actuating by the sensed values as the existing primary controller. However, the functionally duplicated controller is isolated from network, and is made on different hardware such as field programmable gate array

(FPGA) and run on a completely different software environment, hardware-coded without traditional software coding.

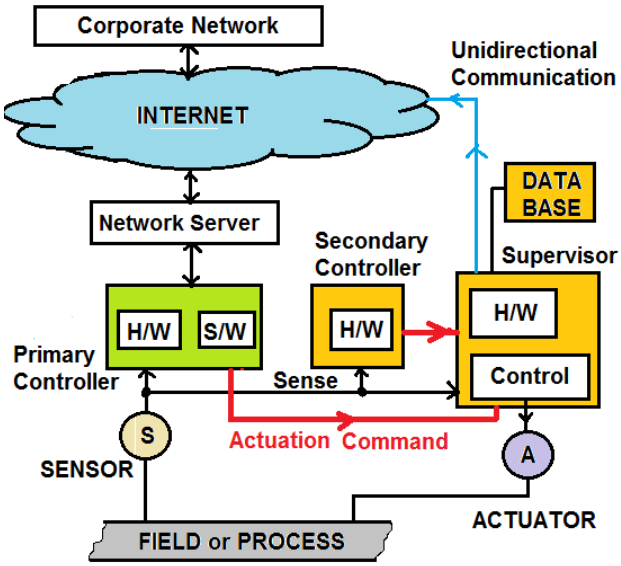

Figure 2. New Architecture for Control Systems

Additionally, the new architecture contains another hardwire device which works as a supervisor of the two functionally identical controllers. In the actuation, there is a slight variation in the new architecture in that both primary and secondary controllers do not directly determine the actuation but each sends actuation command to the supervisor. Therefore, the supervisor is in charge of the eventual control of the actuator. As in the existing connected control system, the primary controller is connected to the communications network server, but the secondary, functionally duplicated controller is isolated from the communication network.

The supervisor is also separated from the communication network, and reads the actuation commands of both controllers and decides if either one is erroneous or not, by conferring with a database which contains data readings previously collected at the sensors and the corresponding actuations over an extended period of operational hours. Under regular operating conditions, there should be near perfect correlation for a given sensed value between the actuation commands generated by the two controllers and the cached actuation in the database.

In the event the supervisor finds discrepancy between the actuation command of the primary controller and the database, for example, the supervisor transfers the actuation command from the secondary controller to the actuator, and at the same time, sends a message of possible cyber infiltration and compromise via a dedicated, unidirectional network to alert the management personnel of the enterprise network. The important and distinct advantage of using unidirectional network connection is that this new system at no point is required to receive and act on requests [6]. Hence, the integrity of alerting is preserved and the possibility of communication related intrusions such as Denial of Service (DoS) attacks is inherently prohibited.

In case the actuation command from the secondary, duplicated controller is found to be wrong, the supervisor keeps the command from the primary controller as the primary actor and sends alert of possible random hardware failure or hidden bugs in the hard-wired code of the secondary controller.

3.3 Qualitative Evaluation

Before we test the new approach, let's do qualitative assessment on its claimed strength against cyber-attacks under three typical instances. The scenarios of attacks are described using Figure 2.

First, we consider the presence of a common computer virus which gains entry into the system through the communication network server via a negligent control systems engineer. Under this circumstance, the operation of the primary controller becomes compromised; however, the secondary controller, by way of design diversity and isolation from network, remains unaffected. Even in the instance of viruses that have the ability to propagate across networks, the difference in software coding methodology between the two controllers grants mutual exclusion in the software attacks, eliminating the threat of common-cause virus infections.

Second, we consider a man-in-the-middle attack. This scenario involves attacks in which access credentials are mined from unsuspecting parties. In this case, for a control system, it is difficult to determine if the system is under attack because the information used to gain unauthorized access to the system is indeed legitimate. Therefore, changes can be made to the primary controller as if authorized firmware update without any intrusion detectors being set off. But even in this compromised state, by virtue of the comparison check that occurs continuously at the supervisor, any changes or discrepancies generated by the intruder are detected, and controlling of the actuator is committed to the isolated thus unaffected secondary controller, keeping the normal control function intact.

Third, we consider a scenario of common-mode hardware failure and software bugs in the primary controller. Hardware failures in this context refer to incidents such as purposeful or accidental physical damage and hardware component faults. Under this scenario, the secondary controller functions not susceptible to the common-mode hardware failure. This ensures that the sensing-actuation operation is maintained and does fail-operate until the proper repair and replacement procedures for the primary controller can be carried out. While the probability of simultaneous failure of both the primary and

secondary controllers of the proposed system exists, it is theoretical and very small.

The qualitative assessment on the above three scenarios demonstrates that the proposed new control system can withstand and fail-operate even under a mode of attack employed by a Stuxnet-like worm in its various iterations. The Stuxnet worm is a program that was developed to target specific industrial software on a specific brand of equipment in a plant [3]. This type of specialized attack is hard to defend because it relies on targeting and exploiting certain vulnerabilities in the operating system. Fortunately, the design diversity afforded by the new system structure acts as a functional safeguard. Having both controllers run on very different software and hardware architectures ensures that whatever illicit alteration or firmware update is done is limited to the primary controller, and such infiltration is captured by the supervisor and alerted, instead of being blinded until more disastrous harm is done.

The proposed diversity architecture of the connected control system upgrades the existing system to a multi-tiered, cooperative system in which desired control functions are kept intact all the time, fulfilling the robustness and fail-operate requirement of the systems that handle critical tasks of, for example, smart-grid control or vehicle acceleration control.

To test the feasibility of the proposed architectural solution, laboratory hardware experimentation is conducted. In the experimentation, the primary controller is represented by a microcontroller, the isolated secondary, duplicated controller by a FPGA board, and the supervisor also by a microcontroller. Then, the two models, the existing control system and the new diversified control system, are subjected to the same attack condition and the corresponding actuator responses are compared.

4 Validation in Experimentation

This small-scale hardware experimentation of the existing and the proposed new control systems is not to test on industry-grade real hardware or scaled-down replica, but to do on a limited scale of logically equivalent hardware components, which are not directly relatable to industry specific components in use. Also, other unrelated specifics such as response times are not considered because they would largely be dependent on the real components that would be used if this new architectural approach is adopted.

4.1 Experimentation Setup

The network environment and the hardware components used in the experimentation are explained using the schematics of Figure 3. As illustrated in the schematic, the primary controller is represented by an Arduino microcontroller [7]. The secondary controller is realized by a Nexys 2 Spartan-3E FPGA board [8]. The supervisor is represented by another Arduino.

The primary controller is connected to the Internet via a network server (which is realized with an Internet-connected laptop (with IP address of 10.232.100.114 for the experiment), and the supervisor is directly connected to the Internet via an Arduino Ethernet Shield [9]. To represent an upper level control and management system A Twitter account, "ArduinoHU," is made to simulate the unidirectional message transmission upon a cyber-incident.

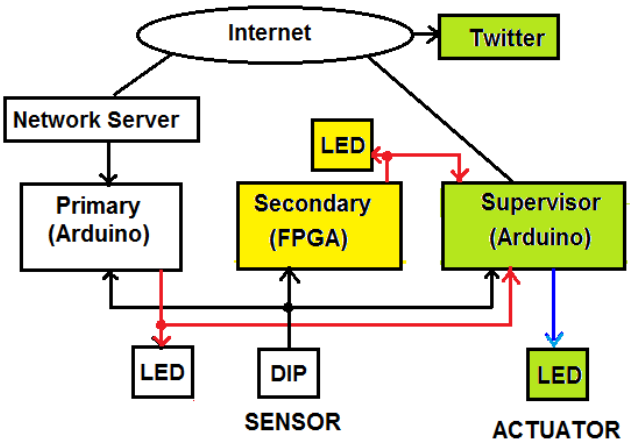

Figure 3. Schematics of Hardware Experimentation.

The sensor is represented by a DIP switch, the position of each toggle of which can simulate various conditions of the application field. The DIP is directly connected to both primary and secondary controllers as well as to the supervisor. The digital command for actuation from each controller is connected to the supervisor and visually indicated by an LED of its own. The supervisor issues an actuation signal after comparing the commands of the controllers with the database. The actuator is implemented by an LED (attached to a simple magnetic relay which is signaled by the supervisor), whose ON/OFF status indicates the actuation state of the control system.

As for software, a simple code is programmed for the primary controller so that it reads values from the DIP switch and sends out corresponding actuation commands based on the pre-set threshold value. The secondary controller is hard-wire coded to perform the same control function. The supervisor is coded to take in two outputs and compare them with a database of past sensor readings and respective actuations which is nothing but a simple data table embedded in the code. Based on the schematic, the lab hardware setup has been implemented.

Figure 4 depicts the experimentation setup disposed on a breadboard. As explained above, the laptop in the figure represents the communication network server for the control system.

For testing for the existing control system, the

communication network server, Internet-connected laptop, and the primary controller and the DIP switch are utilized, which are placed in the left half side of the experiment setup of Figure 4. On the other hand, testing for the new control system architecture includes the entire experimentation setup.

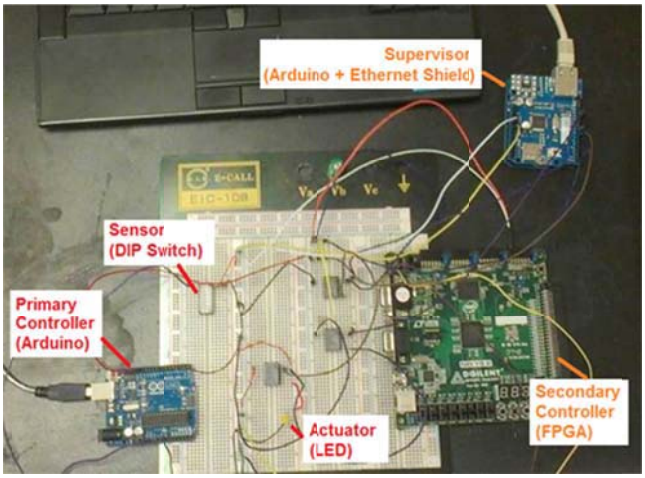

Figure 4. The Lab Hardware Experimentation Setup for the Diversified Architecture

In both tests, it is assumed that the attack is made through the virtual private network of the control system after obtaining the credentials of the manager's computer (with IP address of 10.23.100.19 for the experiment) by means of Trojan virus or a key logger and that the hacker has access to a manager's computer which is connected to the communication network server. Both the manager's computer and the network server laptop (with IP address 10.232.100.114 as previously indicated) used in the experimentation run on Microsoft Windows operating system.

4.2 Staged Attack on Existing System

Once the hacker has the credentials to the manager's computer, the hacker easily connects, using the Remote Desktop Tool of the Microsoft Windows, to the remote network server. After the VPN connection is successfully done, the following message is popped-up:

```
Ravi is now connected.
Assigned IP: 192.168.200.14
```

Then in the Remote Desktop Connection window, the hacker inputs the IP address of the network server to which the primary controller is connected:

```
Computer: 10.232.100.114
```

```
User name: CNE
```

The "CNE" above is the login user name for the laptop. After "Connect" is clicked, then the credentials for the server appears:

```
Enter your credentials
There credential will be used to
connect to 10.23.100.114
CNE:[Password]
```

After the password is entered, the hacker has now access to the application that can update and revise the software code and upload it to the primary controller.

In this particular experiment, the hacker changes an Arduino code and uploads the rogue code to the primary controller which frequently changes the threshold value for actuation. And in the experiment, it indeed results in producing blinking LED every one second while the DIP switch positions are remained intact. This experiment illustrates the exploitable vulnerability of the existing connected control systems and the resulting possible safety-threatening situations in, for example, utility system control or connected vehicle steering wheel control.

4.3 Staged Attack on New Connected System

The same attack is staged for the experimentation setup representing a proposed new control system. Again, sensing is simulated with the positions of the DIP switch and a certain threshold value is coded and hard-wired in to the primary and the secondary controllers, respectively. The supervisor now receives two actuation commands from both controllers, and compares them against a look-up table embedded in its software code. We keep the same assumption that the hacker enters the network and places the same corrupted code in the primary controller. However, we expect, since the secondary controller and the supervisor are not connected to the network, the new system does not suffer from the attack because the secondary controller keeps its operational logic in its hare-wired code.

Therefore, while the primary controller produces and sends erratic ever-changing actuation commands to the supervisor, manifesting with flashing LED of its own, the secondary controller sends consistent actuator commands to the supervisor correctly on DIP positions. Upon the commands from two controllers, the supervisor compares the two against the normal operation history from its database, and selects the secondary controller's command to govern the actuator, manifesting the state of the main LED the same as that of LED of the secondary controller.

As soon as a discrepancy in actuation commands is captured by the supervisor, it sends a twitter message to the "ArduinoHU" account, stating that the primary controller

has malfunctioned and alerts the upper level management of cyber infiltration and of immediate fix and repair of the compromised component.

The following lines show a few relevant parts of the code (with line numbers not same as in the actual code) for twitting message to the "ArduinoHU" account:

```
1 Char thingSpeakAddress[]="api.thingspeak.com";
2 String thingtweetAPIKey="MP98YD0EM36M6BE7";
3 EthernetClient client;
4 updateTwitterStatus1("Relay1 down");
5 updateTwitterStatus2("Relay2 down"
6 void updateTwitterStatus (String tsData) {
7 if (client.connect(thingSpeakAddress,80)) {
8 tsData= "api_key="+thingtweetAPIKey+"&status="+tsData;
9 client.print("POST /apps/thingtweet/1/statuses/
update HTTP/1.1\n");
10 client.print ("Host: api.thingspeak.com\n");
11 client.print ("Connection: close\n");
```

In the Twitter account, over the experiment, the alert message has been registered (1 minute before the account is accessed) as listed below along with other previous usual messages:

```
Tweets
ArduinoHU @ArdunioHuU                1m
Realy1 down
Expand
ArduinoHU @ArdunioHuU                17 Apr
Rishi
Expand
ArduinoHU @ArdunioHuU                10 Apr
Ravi
Expand
```

Hence, even under the compromised situation in the primary controller (which is the same as the existing control system), the intended functions would survive and there would be no disruption of service to customers. This lab experiment demonstrates that the proposed new control system can survive and fail-operate cyber-attacks.

However, a minor problem is noticed in simulating the unidirectional message alert from the supervisor to the upper management via Twitter message. Under this setup, it is possible that the message being sent to the Twitter account is captured and replaced with false message. Even though the false message would not harm the control system's operation, the upper management would not be alerted of the cyber infiltration to the remote control systems. It is hoped that, in real application of the proposed control system, the suggested unidirectional fiber optic network would do the intended function properly.

5 Conclusions

Inevitable side effects of the connected devices are cyber threats and attacks, skills and tactics of which are constantly evolving. Even with numerous countermeasures developed and deployed, new attacks seem to materialize as soon as old ones are solved or patched. To cope with the impact and consequence of the service interruption caused by cyber-attacks to safety-critical connected control systems, new system architecture was proposed. The new control system was realized with the hardware/software diversification principle and the supervised operation accompanied with unidirectional communication. We tested the proposed system in the lab hardware experimentation, and demonstrated its validity and the survival potential under cyber-attacks. This new control system architecture would assure cyber-robustness and resilience even under compromised situations.

References

[1] Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-in-Depth Strategies, Department of Homeland Security. September 2014,

[2] D. Kushner, The Real Story of Stuxnet, *IEEE Spectrum*, Available: http://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet. .

[3] K. Stouffer, J. Falco and K. Scarfone, Guide to Industrial Control Systems (ICS) Security, 2nd revisions, June 2015. Available :http://nvlpubs .nist.gov/nistpubs/ SpecialPublications/NIST.SP.800-82r1.pdf.

[4] R. Langner, *Robust Control System Networks*, NY, NY: Momentum Press, 2012.

[5]Charles Kim and Peter Keiller, "A Course Development Project for Hardware Diversity and Software Reliability Education for Digital Instrumentation and Control of Nuclear Power Plant," 2011 Conference on Nuclear Training and Education, Feb 6-9. 2011. Jacksonville, FL.

[6] Andrew Ginter, "Waterfall Security Solutions – Prepared Remarks", FERC 2016 Supply Chain Integrity Technical Conference, Washington, DC. January 28, 2016

[7] "Arduino," Available: http://www.arduino.cc/. (Access: July 2016)

[8] "Nexys 2 Spartan-3E FPGA Trainer Board," Available: https://reference.digilentinc.com/nexys:nexys2: refmanual

[9] "Arduino Ethernet Shield" Available: https://www.arduino.cc/en/Main/ ArduinoEthernetShield (Access: July 2016)