CYBER RESILIENT HARDWARE CONTROLLER

P. PEDERSON
G. FINK
N. CHRISTMAN
Pacific Northwest National Laboratory
Richland, Washington USA
E-mail: perry.pederson@pnnl.gov

Abstract

General-purpose computation has improved the lives of billions around the globe. To include the nuclear industry, critical infrastructures depend upon the ubiquitous use of networked computerized devices. Due to the commodity components used in their design, these devices have excess capabilities and processing capacity that provide a larger attack surface. To reduce this attack surface, the authors describe a prototype of a hardware-only controller, a deterministic system, developed by the authors, that performs its intended function, and nothing else. The paper posits this device as a viable replacement for software programmable controllers and describes future research paths.

1. INTRODUCTION

Cyber-physical (cybernetic) control systems in our national critical infrastructures are vulnerable to Internet-borne attacks against the software they run. While society needs all the intended functionality of the software in these cybernetic systems, at least one source of vulnerabilities can be eliminated entirely—the software itself.

Software has become the primary means of intelligent process control (cybernetics) in industry [1] for many reasons to include flexibility and software also offers over-the-Internet maintenance updates that greatly reduce cost. The resulting economic incentives have led to digitizing and automating everything possible, even our critical infrastructures upon which our lives depend. The most direct approach to build a software-based controller is to employ commercial off-the-shelf, general-purpose computers and opensource software. Unfortunately, the convenience of networking and the monoculture created by commodity hardware have made software the great Achilles' heel of any software-based device, especially for control systems and simple devices like those described as the Internet of Things (IoT). The commodity hardware is far more capable than what is typically needed to control the physical process in question. This extra capacity is attractive to attackers, as learned from the Mirai botnet that produced the largest distributed denial of service (DDoS) attacks thus far recorded [2]. The Mirai botnet (a network of private computers infected with malicious software and controlled as a group without the owners' knowledge) overwrote the software embedded in the controllers of billions of IoT devices, like digital video recorders, producing a massive DDoS attack against Internet routing services and shutting down large portions of the Internet for a period.

Although cyber-attacks against industrial controllers may not hit the headlines as frequently as ransomware attacks, the potential for a catastrophic result is much more likely. A case in point is the first well documented case of an attack on industrial control systems at the Natanz nuclear facility. According to Langner [3], Stuxnet was the first documented case of a direct cyber-attack on a controller and resulted in hardware failure. The potential for loss of life or damage to equipment or the environment are ever present when hackers turn their sights to industrial control systems.

2. BACKGROUND

This paper employs the term 'cybernetic' that was first defined in the 1940s by Norbert Wiener [4], who studied human and machine control mechanisms and envisioned some of the issues that society are now grappling with, namely cyber security.

At about the same time that Wiener was developing his ideas on cybernetics, John von Neumann was working on a report that has come to be known as the von Neumann architecture [5]. The basic structure proposed in his writing contains memory, a processing unit, and a control unit and remains as the basic architecture of every computer, tablet, smart phone, or industrial controller in the world today. There are two edges to this sword. Because of the flexible Von Neumann architecture, systems can be repurposed to nefarious ends with new software, in other words malware. The machines will simply do what the software tells them to do. According to Dan Geer, society has yet to learn the lessons of connecting everything to everything [6].

3. CONTROLLER DESIGN

The goal of this effort was to reduce cyber risk following the suggested strategy by well-known thought leaders such as the previous Secretary of the Navy, Richard Danzig [7]. Our goal is also aligned with some core issues currently being debated at the policy level in the U.S. Congress [8] to enhance safety and security through a systematic reduction in the complexity of control systems.

Our design efforts also leveraged work on nuclear safety systems [9] related to using hardware to execute safety logic. This project used a Field Programmable Gate Array (FPGA), but only as an interim step for proof of concept. While this idea of using an FPGA in and of itself is not new or unique, the path forward is to replace the FPGA with a 3D printed circuit board. Ultimately, this could reduce the cost and cycle-time of the typical Application-Specific Integrated Circuit design process and put this capability into the hands of the typical process engineer.

Our effort demonstrated the feasibility of developing a hardware-only controller that due to its inherent design, cannot be repurposed remotely because it will not contain any software. Importantly, this solution is not turning back the clock to old analogue solutions. Instead, a modern digital controller was built without the inherent vulnerability and remote alterability of software-based controllers. This approach to building a controller will require physical access to each one to override their physically programmed functions. They will be provably (not just theoretically) un-hackable without direct physical access. The physical access requirement means that system-wide changes cannot simply be rolled out to a national enterprise from a remote location. While it is true that the cost of updating hardware-based systems would be higher on a per-controller basis, most industrial control systems are not supposed to change function or be updated for decades at a time. Arguably, the reason most security updates or patches are needed today can be traced to the vulnerabilities discovered in the software post deployment. With a hardware-only controller, the testing of all possible states is possible and security updates will no longer be needed. For critical infrastructure, the security afforded by this inflexibility will be a major strength that outweighs the cost incurred by the rare need for system updates.

At a high level (Fig. 1), the ladder logic diagram for a notional safety alarm system was taken from the Programmable Logic Controller (PLC) Manual [10] website and the design was implemented in Verilog Hardware Definition Language using the Intel Quartus development platform, and then the logic was integrated with an open source User Datagram Protocol (UDP) firmware project built by Alex Forencich [11].

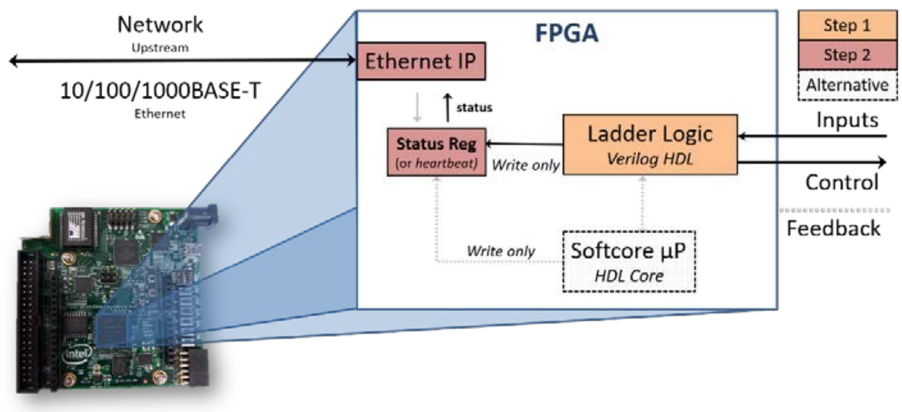

Fig. 1: High level design for hardware-only controller

The project is split into two components: the FPGA firmware and the PC host application software. The firmware was created and synthesized using Intel Quartus Prime version 17.0.0 Build 595 04/25/2017 SJ Standard Edition. The target device for this project is an Intel Cyclone 10 low-cost, low-power development kit and the Joint Test Action Group Indirect Configuration bitstream file is properly configured to program the Quad Serial Peripheral Interface flash on this development kit.

Given the physical architecture of an FPGA, the UDP firmware employed for this project is application specific and, as such, does not include any software; thus, meeting the requirement of having network connectivity without software. The PLC specific firmware was solely written at Pacific Northwest National Laboratory (PNNL). The host software is a C# windows form application. The software was created and compiled using Microsoft Visual Studio 2017 Community Edition version 15.5.7.

The hardware-only controller and the user interface (Fig. 2) was developed to emulate an alarm system that might be found in an industrial environment. In this alarm there are four (4) sensor inputs to protect the factory, so that in case of any signals on any of these inputs, it will give a certain alarm. The controller responds to the sensor input signals processed according to the ladder logic based on the following requirements:

- If only one of the inputs (sensors) is ON, nothing will happen.
- If two of the inputs are ON, the Red Pilot Light will be activated.
- If three of the inputs are ON at the same time, an alarm would be triggered.
- If all four inputs are ON together, the fire department is called because fire will erupt.

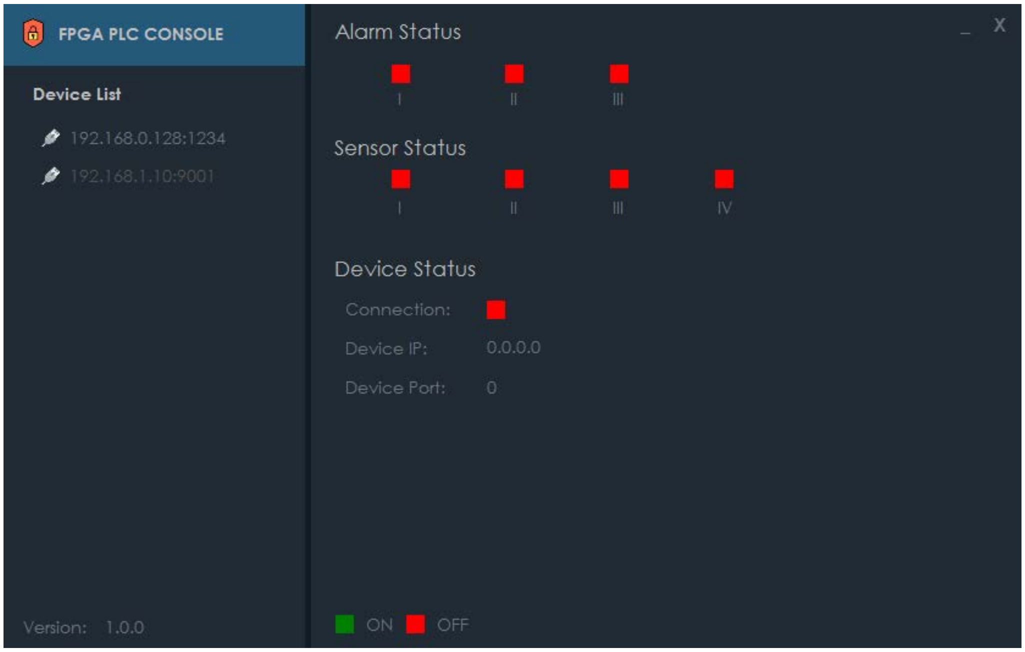

Fig. 2: Host application user interface

4. RESULTS AND EVALUATION

The project met its established goals and held a live demo at PNNL in 2018. We demonstrated how it is possible to control process logic using a hardware-only controller. Thus, this project established that FPGAs can be made functionally equivalent to PLCs except that the FPGA solution does not have software that can be compromised by a network-borne cyber-attack. However, has noted previously, using an FPGA is just an interim step. Ultimately, the goal is to replace the FPGA with a 3D printed circuit board capable of executing the same logic.

5. DISCUSSION

Software-based systems embedded in critical infrastructures are difficult to protect. Various protection approaches to fixing or mitigating the vulnerabilities induced by software include enhanced operator cyber-awareness, air gapping, intelligent symbiotic programs, and encrypted central processing units (CPUs). Enhanced awareness does not fix the vulnerabilities in these systems but instead purports to catch intrusions more quickly by making humans better at detection. Instead, the system as proposed herein significantly reduces the risk of being hacked in the first place.

Air gapping is a way to isolate vulnerable critical systems from Internet connections by placing them on entirely separate, internal-only networks. Unfortunately, in practice an air gap is just a high latency connection. To stay up to date, to send status, or to receive new instructions, air-gapped systems must connect to systems that are in some way connected to the outside world such as a maintenance technician’s laptop or USB memory stick. Thus, air gapping helps, but does not completely solve the problem.

Red Balloon Security has created an intelligent “symbiote” software program that chops the legacy firmware into pieces and interleaves itself among them [12]. It executes in parallel and constantly checks the state of its own execution and that of the programs to be protected. If either is corrupted, it raises an alarm. However, utilities and other companies have been slow to adopt this approach because the firmware must be separately generated for each

CPU/functional combination. Additionally, it appears the solution is difficult to prove reliable because it rearranges legacy software, making existing certifications void and often voiding warrantees.

Encrypted CPUs use an FPGA in a similar way to what is described herein to solve the problem by making each CPU's instruction set unique. Software running on these systems must be encrypted with a symmetric key encoded in the FPGA, which decrypts and executes the program. Thus, injected code, which is not properly encrypted, cannot run on the machine. This process introduces additional levels of complexity, such as key management, which is itself another problem.

The most likely first application of a hardware-only controller is safety-critical systems, because of their general simplicity and static nature. However, over time, as tools for quickly converting normal software to hardware-only implementations becomes more available, these non-software solutions could prove viable for non-safety-critical applications such as the IoT.

6. CONCLUSIONS

While this project demonstrated feasibility, additional research will be required to address market viability by exploring the limitations of current 3D printing technology and estimating adoption cost. We will also have to develop general tools and methods to increase market penetration. The team plans to create additional prototypes that demonstrate a wide range of possible commercial applications, such as a video camera or network controller, to make transition to practice more likely.

REFERENCES

[1] ANDRESSEEN, M, 'Why Software Is Eating The World', The Wall Street Journal, 20 August, 2011, https:/www.wsj.com/articles/SB10001424053111903480904576512250915629460.

[2] U.S. DEPARTMENT OF HOMELAND SECURITY COMPUTER EMERGENCY RESPONSE TEAM, Alert (TA16-288A) Heightened DDoS Threat Posed by Mirai and Other Botnets, 2016, https:/www.us-cert.gov/ncas/alerts/TA16-288A.

[3] LANGNER, R., Stuxnet: dissecting a cyberwarfare weapon, IEEE Security and Privacy, vol. 9, no. 3, pp. 49-51. 2011.

[4] WIENER, N., Cybernetics: Or Control and Communication in the Animal and the Machine, Hermann & Cie, Paris, MIT Press, Cambridge Massachusetts, 1948.

[5] ASPRAY, W, John von Neumann and the Origins of Modern Computing, MIT Press, Cambridge Massachusetts, 1990.

[6] GEER, D., Cybersecurity as Realpolitik - keynote, Black Hat, viewed 21 June 2019, https:/catless.ncl.ac.uk/Risks/28/15.

[7] DANZIG, R., Surviving on a Diet of Poisoned Fruit Reducing the National Security Risks of America's Cyber Dependencies, Center for a New American Security, 2014, https:/s3.amazonaws.com/files.cnas.org/documents/CNAS_PoisonedFruit_Danzig.pdf?mtime=20161010215746.

[8] U.S. CONGRESS, S.79 - Securing Energy Infrastructure Act 2018, viewed 21 June 2019, https:/www.congress.gov/bill/115th-congress/senate-bill/79.

[9] REBSTOCK, P. JR., A Digital Safety Actuation System for Nuclear Power Plants: Suppressing the Perils, Meeting the Promise, Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, 2009.

[10] PLC MANUAL, viewed 28 June 2019, http:/www.plcmanual.com/programming-examples-v.

[11] FORENCICH, A., Verilog-ethernet, viewed 28 June 2019, https:/github.com/alexforencich/verilog-ethernet.

[12] CUI, A., SALVATORE, S., Defending embedded systems with software symbiotes, International Workshop on Recent Advances in Intrusion Detection, pp. 358-377, Springer, Berlin, Heidelberg, 2011.