Hardware Components in Cyber Security Education

Dan Chia-Tien Lo[1], Max North[2], Sarah North[3]
[1]Department of Computer Science and Software Engineering
Southern Polytechnic State University
[2]Department of Business Administration
Southern Polytechnic State University
[3]Department of Computer Science
Kennesaw State University, USA
{clo, mnorth1}@spsu.edu, snorth@kennesaw.edu

ABSTRACT: *Hardware components have been designated as required academic content for colleges to be recognized as a center of academic excellence in cyber operations by the National Security Agency (NSA). To meet the hardware requirement, computer science and information technology programs must cover hardware concepts and design skills, topics which are less emphasized in existing programs. This paper describes a new pedagogical model for hardware based on network intrusion detection taught at college and graduate levels in a National Center of Academic Excellence in Information Assurance Education Program (CAE/IAE). The curriculum focuses on the fundamental concepts of network intrusion detection mechanisms, network traffic analysis, rule-based detection logic, system configuration, and basic hardware design and experiments. This new course enriches students with the latest developments in computer security and hands-on projects to better prepare them for their information security and assurance careers.*

Keywords: Network Intrusion Detection, Hardware, FPGAs, Computer Science Education

Received: 30 November 2013, Revised 5 January 2014, Accepted 12 January 2014

1. Introduction

Cyber security, or Information Security and Assurance, has become a crucial discipline over the last decade, and a top concern to the nation. Meanwhile, the most popular mobile platforms have attracted an enormous amount of security breaches as global mobile payment transactions are predicted to hit $1 trillion by 2015 [1]. The growing importance of cyber security has resulted in a new knowledge area of 9 core hours of Information Assurance and Security, according to the IEEE/ACM Curricula Guidelines CS 2013 [2]. However, cyber security remains less emphasized in computing related programs, and hardware educational components are fading in Computer Science (CS) or Information Technology (IT) programs.

CS and IT curricula grow and change rapidly and continuously, owing to technology advancement. This growth causes an inevitable curricular revision every four years, as IEEE/ACM joint curriculum recommendation committee suggests. Without increasing the gross credit hours of a program, the fast expansion of the discipline leaves little room for adding new courses that address emerging areas such cyber security or many-core programming, and meanwhile compresses existing courses such as Computer Architecture and Organization. One workaround will be either integrating new knowledge areas into existing courses,

or bundling several knowledge areas in one course.

Based on the NSA's academic requirements for designation as a center of academic excellence in cyber operations [3], a program must cover the following hardware knowledge areas:

• Assembly language programming (x86, ARM, MIPS or PowerPC),

• Software reverse engineering (need programming skills in assembly languages),

• Programmable Hardware design languages (schematic and Very large scale integrated circuit Hardware Description Language (VHDL)),

• Field Programmable Gate Arrays (FPGA) Design,

• Computer Architecture,

• Microcontroller Design,

• Software security analysis (binary code analysis), and

• Hardware reverse engineering.

Most of the above hardware knowledge areas involve basic understanding of how programs are executed on a bare machine, as opposed to writing a working program on a "*black box*". In fact, the "*black box*" serves the root of cyber security as delineated in a recent publication from National Institute of Standards and Technology (NIST) [4]. According to NIST, hardware roots of trust (RoTs) are preferred over software RoTs due to their immutability, smaller attack surface, and more reliable behaviors.

One cyber security issue closely related to hardware engineering is network intrusion, which deals with network packet sniffing, analyzing, protocols, and configurations. Network intrusion detection systems (NIDS) have been widely deployed, either in a complete hardware and software system or in a software solution such as Cisco intrusion detection system appliances, Snort [6] and Bro NIDS [7]. In order to deploy and maintain a NIDS in a network environment, one must fully understand the fundamental concepts of each component in the system, such as routers, switches, packet capture devices, network layers, Internet Protocols [IPs], Internet Control Message Protocol (ICMP), Ethernet frames, and the like. To maintain a NIDS, a network administrator will have to be able to write and read rules used for intrusion detection, to understand the pros and cons of a network configuration with regard to vulnerability, to analyze the alerted logs generated and take appropriate actions, and to diagnose anomalous or malicious network traffic.

Therefore, to complement the existing computer science and information technology programs, this new hardware component provides students with the ins and outs of network intrusion detection systems such as Snort, the fundamentals of IP networking, system configurations, and basic hardware design. A hands-on term project is designed for students to exercise the aforementioned fundamentals through a small scale instruction detection system on FPGAs, which allow them to develop their hardware and software including a traffic generator.

2. Course Objectives

This course introduces students to the latest developments in hardware support for network security. In addition to network security fundamentals, research projects in this field are presented. Students will be exposed to hardware and software design and an implementation for a complete network intrusion detection system. The expected course outcomes are listed as follows:

• Demonstrate the basic hardware design related to cyber security (schematic design and VHDL design)

• Understand a network intrusion detection system

• Demonstrate familiarity with a variety of intrusion detection rules

• Analyze and design network traffic such as IP, ICMP, Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) packets

• Design and develop hardware for intrusion detection

• Configure and maintain a network intrusion detection system

The course outcomes are designed to be measurable. For example, to measure student's ability to analyze and design network

traffic such as IP, ICMP, TCP, and UDP packets, a traffic generation assignment is created, in which students must learn the formats of network packets and protocols such as TCP 3-way handshaking in order to transmit or decode packets.

3. Covered Topics

The topics taught since the Spring 2010 semester at Southern Polytechnic State University are listed here:

#	Topic
1	Introduction, course organization.Network Security, Intrusion Detection
2	**Project Description** **A Network Intrusion Detection system:** SnortSystem programming tools and techniques
3	**Hardware Platform:** An Embedded SystemEmbedded System Design Tools
4	Traffic Analysis, Generation, and Recognition
5	**Packet Manipulation Libraries:** Libnet, Pcap, Tcpdump
6	**Network Protocols:** IP, TCP/UDP, ICMP, Address Resolution Protocol (ARP)
7	**Hardware Design:** VHDL/Verilog Basics
8	User IP Design and Integration
9	Device Drivers, I/O Controls, Direct Memory Access (DMA)
10	Performance Evaluation
11	Perl Compatible Regular Expression (PCRE) Rules
12	Automatic Worm Fingerprinting
13	On-going Research Projects

4. Project

The goal of the project is to familiarize students with basic hardware and software design as it relates to network traffic analysis. In order not to interfere with normal network traffic, we adopt Xilinx FPGA development boards (XUPV5-LX110T) for the development platform (Figure 1), which includes a network port, and a soft-core network controller. This platform allows students to build their intrusion detection hardware and software from ground up. This project includes 4 parts: Binary adder, decoder, network traffic generation, and receiving network traffic as detailed in the following sections.

Figure 1. Xilinx FPGA Development Board (XUPV5-LX110T)

4.1 Binary Adder

The purpose of this assignment is to let students become familiar with the schematic design tool in the Xilinx integrated design environment (ISE), an IDE for both hardware and software. Since the binary adder has been taught in prerequisites, students only need to exercise the operations of the Xilinx ISE tool. Moreover, they will learn basic VHDL programming (when creating a test bench with stimuli), simulation, and debugging. A free version of the Xilinx ISE software, dubbed Web pack, is sufficient to work on the project.

4.2 Decoder

As reported by many researchers, a significant amount of time is spent in pattern matching in NIDS, and a very possible solution is shift the matching to hardware. In this assignment, students are to design a logic block for 8×256 decoding that has one byte input, and 256 data bits as output using VHDL. This example addresses the need to use hardware description languages (HDLs). Without HDLs, it would be very tedious to design such a decoder in schematic design. This decoder is the fundamental component for a hardware matcher mentioned by Hutchings et al. in their hardware assisted network intrusion detection using reconfigurable FPGAs [5]. The decoder logic will assert an output bit for a specific input byte. For example, the 65^{th} output bit is asserted if the input byte is "*A*", because the ASCII code of "*A*" is 0×41. By connecting the outputs bits accordingly, a hardware pattern matcher would be constructed systematically.

4.3 Insecure Telnet

Telnet exposes a user's ID and password in a network. This project attempts to address this security concern and requires students to design a program in any programming or script language to extract a user's telnet ID and password from a log of dumped packets. The program must be executable in Linux or Windows.

To do list:

1. Follow the unsecure telnet video to create a log file of packets

2. Make sure that the packets containing the user's ID and password are logged

3. Develop a program in any programming language or any script that reads text from the standard input, processes the information, and outputs the user's ID and Password via standard output. The command line for running the program must be

```
$sudo snort -dv -r
snort.log.1234567891
| myPassExtractor
```

where the program name is myPassExtractor, and snort.log.1234567891 is the packet log file.

4.4 Traffic Generation

Network traffic is normal for most of the time. In order to test components in NIDS, it is inevitable to artificially generate anomalous traffic. In this assignment, students will implement a program, running on the XUPV5-LX110T board, which will send out an Ethernet frame to a host via a Tri-Mode Ethernet Media Access Controller (TEMAC). The TCPDump running on the host will be used to capture the frame. The captured packets are then compared to those sent by the sender.

Hardware: The EDK Base System Builder (BSB) is used to build a hardware configuration with a TEMAC. We provide a working version for student's reference, So they may start working readily.

Software: Students will use Xilinx EDK 11.1 to code their programs. They can use any computer to create a hardware project, in which a software project is associated. With EDK, compile, simulation, and debug programs will be performed seamlessly. Once the system is simulated correctly, students will physically go to the lab to program their hardware and to test their systems.

Ethernet Frame: The frame to be sent will carry an IP packet with a UDP datagram. Therefore, it provides a place for students to learn and validate all headers and payloads. In the payloads, they may put a pattern in the UDP datagram to be identified later. Also, IP headers will have to be filled.

TCPDump: To analyze network traffic, the widely used free tool TCPDump is employed to capture and verify the packet.

Students can use Microsoft Visual Studio or Cygwin to development a traffic generator using the Pcap library. The traffic generator should generate multi-protocol packets such as TCP, UDP, ICMP, etc. An example of the traffic generator is illustrated

as follows:

```
$tg – p tcp –s 100 –i 50 –c "AbC " –srcip
192.168.1.100 –dstip 192.168.2.100 –
srcmac 0:1:2:3:4:5 –dstmac 0:a:b:c:d:e.
```

This command generates 50 TCP packets of size 100, each of which contain "*AbC*" in their payloads, and corresponding IPs and MACs are set accordingly. Source IP address is designated by `–srcip`, destination IP address is specified by `–dstip`, source MAC address is indicated by `–srcmac`, and destination MAC address is described by `–dstmac`.

4.5 Receiving Network Traffic

In this assignment, students will implement a program, running on the XUPV5-LX110T board, which receives any Ethernet frames on the subnet using a polled mode. In order to receive any IP packets, the Tri-Mode Ethernet Media Access Controller (TEMAC) has to be in promiscuous mode. Otherwise, only the packets targeted at the host may capture them be captured. Students will run their traffic generators, and then run the TCPDump on the host to capture the frames generated from the traffic generators.

4.6 Final Project: Hardware Based Network Intrusion Detection

Finally, students will implement a hardware/software co-designed network intrusion detection system by assembling all components developed as a group project as illustrated in Figure 2. They will attach a hardware matcher that recognizes "*SPSU*" to the Micro blaze CPU, a soft core processor developed by Xilinx, via a processor local bus (PLB). The whole system will be implemented on the XUPV5-LX110T board, which will receive any Ethernet frame on the subnet. The traffic generator will send packets containing "*SPSU*" in their payloads. This is called a signature for an attack. In reality, each attack is associated with a signature. By detecting the signature on a packet, the corresponding attack can be found. Once the hardware matcher detects packets that contain "*SPSU*", it generates an interrupt. In this interrupt, a message will be printed, such as "*found intrusion packets*."

Hardware

Students will use EDK BSB to build a based hardware system with a TEMAC or get a working version from http://cse.spsu.edu/clo/rcl/resource.htm and start from there. Students will need to use "*create and import wizard*" to generate a template hardware matcher (peripheral) with related software components such as device drivers and then design and simulate the matcher with ISE. student's matchers will generate an interrupt if the signature "*SPSU*" is found in a packet.

Software

Students will use EDK 11.1 to code a program running on the board. Students can use any computer to create a hardware project using xps_ll_fifo mode on xps_ll_temac, in which they will create a software project for this assignment. First, the program must be compiled and debugged. Once finished, students will have to physically go to the lab to test the program on the development board. Specifically students will perform the following:

1) Grab packets as before, and send them to the hardware matcher, then

2) Modify interrupt service routines to print a message "*found intrusion packets*" if the interrupt is from the hardware matcher.

Traffic Generation

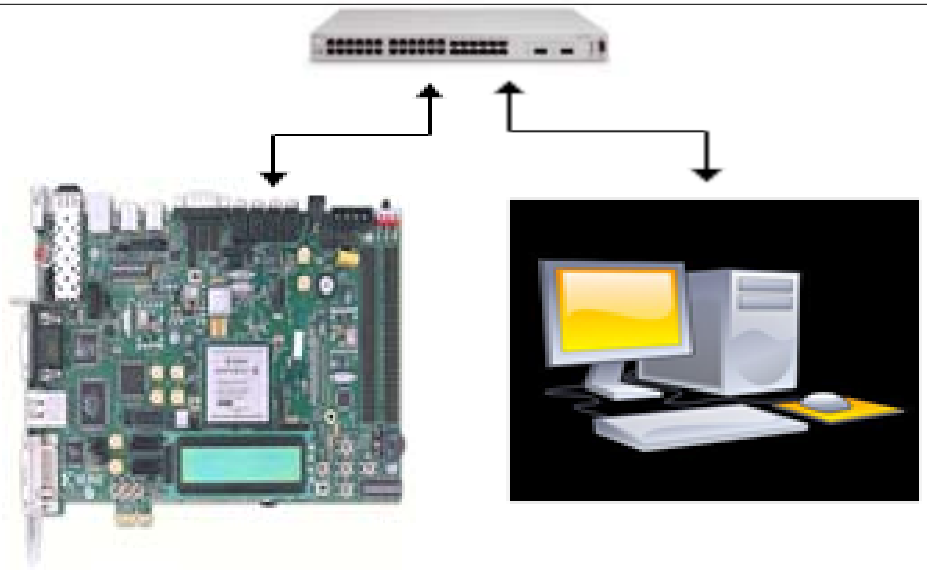
Students need to send packets that contain "*SPSU*" using the traffic generator.

TCPDump

Students might need to use TCPDump to make sure they generate "*correct*" packets.

5. Evaluation

The preliminary experiments on the proposed hardware component curriculum are implemented in two classes, in which student's feedback is very positive. Many students recommend other students take this course because they enjoy what they have learned through the hardware/software co-designed system, as well as development concepts and skills. Especially, students like the hands-on labs with the Xilinx development kit, from which students learn the latest design techniques on systems hardware/software co-design and testing.



XPUV5-LX110T running your program receiving Ethernet frames on the subnet. This hardware will receive and analyze network traffic sent from the host in the right hand side.

Host running EDK 11.1, TCPDump, Hyperterminal, XMD, and tg. This host will generate artificial network traffic and send it to XPUV5-LX110T hardware in the left hand side.

Figure 2. System Configuration for the Final Project

Specifically, students come in with no hardware design skills, and typically walk out with basics of those skills. They will also learn to better analyze network traffic in terms of network packets, and tools used in this course such as TCPDump and Snort. Additionally, students will be able to develop network applications using packet processing libraries such as Pcap.

6. Conclusion and Future Work

With the very positive feedback from our students, we plan to offer the course regularly and continue to improve the material. Owing to a great need in computer security professionals, this new curriculum covers both hardware and software aspects of network intrusion detection, and well-prepares our students in their information security and assurance careers. To encourage advanced studies for working professionals, an online version of this course has also been developed. However, a virtual lab that provides XPUV5-LX110T is under development.

7. Acknowledgments

Our thanks go to the National Security Agency in partial support for allowing us to develop this new hardware component.

References

[1] Report: global mobile payment transactions to hit $1 trillion by 2015, http://www.mobilepaymentstoday.com/article/209617/Report-global-mobile-payment-transactions-to-hit-1-trillion-by-2015, accessed on Jan 30, 2014.

[2] IEEE/ACM Computer Science 2013: Curriculum Guidelines for Undergraduate Programs in Computer Science, http://www.acm.org/education/CS2013-final-report.pdf, December, 2013, accessed on Feb 3, 2014.

[3] Academic requirements for designation as a center for academic excellence in cybersecurity operations, online available at http://www.nsa.gov/academia/nat_cae_cyber_ops/nat_cae_co_requirements.shtml, retrieved in May, 2014.

[4] Chen, L., Franklin, J., Regenscheid, A. (2014). National Institute of Standards and Technology, Guidelines on Hardware Rooted Security in Mobile Devices (Draft), Special publication 800-164. Available online at http://csrc.nist.gov/publications/drafts/800-164/sp800_164_draft.pdf, retrieved in May.

[5] Hutchings, B. L. , Franklin, R., Carver, D. (2002). Assisting network intrusion detection with reconfigurable hardware, *In*: Proc. of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2002), Napa, CA, April, p. 111–120.

[6] Snort-an open source network intrusion prevention and detection system, http://www.snort.org/, accessed on July 7, 2014.

[7] Bro intrusion detection system, http://bro-ids.org, accessed on July 7, 2014.