Design and Implementation of Industrial Control System Emulators

Robert Jaromin, Barry Mullins, Jonathan Butts, Juan Lopez

Chapter 3

DESIGN AND IMPLEMENTATION OF INDUSTRIAL CONTROL SYSTEM EMULATORS

Robert Jaromin, Barry Mullins, Jonathan Butts and Juan Lopez

Abstract The first step to grappling with the security problems that face modern supervisory control and data acquisition (SCADA) systems and industrial control networks is investing in research and education. However, because of the specialized nature of industrial control systems and networks, the cost of even a modest testbed for research or education can quickly get out of hand. Hardware-based testbeds are often not practical due to budgetary constraints, and they do not readily scale to meet educational demands. Software simulations are a cost-effective alternative, but current solutions focus primarily on network aspects, not the implementation of field device and application functionality. This paper describes the design and implementation of a programmable logic controller emulator using VMware. The emulator solution is both cost-effective and scalable. Moreover, it can accurately replicate real-world field device functionality to meet research and educational requirements.

Keywords: Industrial control systems, programmable logic controllers, emulation

1. Introduction

Understanding and mitigating attacks on industrial control systems requires researchers and operators who have extensive theoretical and practical knowledge of industrial processes and control systems. However, the cost of even a modest industrial control system can exceed $100,000, making the investment in security research and education prohibitive for most organizations. Clearly, cost-effective solutions are required to build human capital in industrial control systems security and to analyze, detect and mitigate attacks on industrial control systems that are widely used in the critical infrastructure.

Current solutions for industrial control systems research and education are inadequate in terms of cost and scalability. Full-scale, hardware-based SCADA testbeds are limited to research applications because of their capital, operational and maintenance costs, while smaller educational testbeds often lack the scalability to tackle real-world problems. Meanwhile, computer-based simulations are cost effective and useful for conducting research on certain types of large-scale, network-based systems, but they are inadequate for education and training.

This paper discusses the research and educational requirements associated with industrial control emulators. Also, it presents a novel emulation solution that is attractive in terms of its ability to accurately replicate real-world field device functionality as well as its cost and scalability.

2. Background

Public and private sector entities have made significant investments in experimental industrial control facilities for researching threats and vulnerabilities. Recognizing the need to develop security solutions for industrial control systems, the U.S. Department of Energy spearheaded the creation of the $114 million National SCADA Testbed at Idaho National Laboratory [5]. As America's premier SCADA research facility, the testbed supports the identification and mitigation of vulnerabilities and the development of advanced system architectures for secure and robust SCADA systems. The testbed provides an exceptional environment for investigating real-world SCADA security threats, but similar testbeds are not feasible at academic and research institutions [7]. Mississippi State University, for example, has implemented an educational SCADA network [12], but it is limited to five control devices and two master stations. The environment is practical for educational purposes, but it does not scale up to large industrial applications.

Pure software-based simulations of SCADA networks and hybrid software-hardware configurations have been implemented as alternatives to SCADA equipment. Examples include the Cyber Security Testbed at the University of Illinois [2] and the SCADA Security Testbed at the Royal Melbourne Institute of Technology [14]. These testbeds leverage network simulation frameworks such as the Real-Time Immersive Network Simulation Environment for Network Security Exercises (RINSE) and Emulab [4]. Although they are useful for network-level evaluations, the testbeds do not provide the ability to make fine adjustments to simulate SCADA device features such as firmware uploads and malware propagation

Unlike most simulation efforts, which focus on network-level and/or device-level functionality to replicate real-world industrial control systems, this paper focuses on emulating programmable logic controllers (PLCs) that control the underlying physical processes. PLCs are specialized embedded computers with three core characteristics: (i) the ability to complete electronic circuits and measure electronic signals; (ii) the ability to be programmable to respond to

inputs according to a predefined algorithm or protocol; and (iii) the ability to be networked with other devices.

Emulated industrial control systems combine the advantages of dedicated hardware and cost-effective, scalable software simulations. Emulators achieve these properties by implementing algorithms, protocols and services associated with industrial control systems using flexible programming platforms such as Linux. However, significant technical challenges are associated with emulating PLCs. A single product family, or even a single device, could involve multiple architectures and platforms. To further complicate the task, firmware can be packed or cryptographically encoded so that it can only be decoded by special hardware [9]. Additionally, errors or failures in the operating firmware should, by extension, be replicated in the emulator.

3. Industrial Control System Emulators

This section discusses the applications of industrial control system emulators in research and education, and presents the advantages of emulators over hardware testbeds, software simulations and hybrid hardware-software systems.

3.1 Research Environments

Industrial control emulators provide many of the same advantages as real hardware in a research environment. Accurate emulators not only implement the responses to user queries in the same manner as their hardware counterparts, but they also emulate the operational profile. When protocol responses are emulated accurately, the devices can be used in the same manner as real hardware for research experiments with less cost and reduced setup and maintenance times.

Accurate emulators have several advantages over real hardware devices. In addition to reduced cost, emulators can be more robust because error conditions such as buffer overflows and erroneous firmware uploads do not affect the underlying physical system. While a real PLC might be permanently damaged by such actions, it is a trivial matter to reset an emulated device and continue with the experiments.

3.2 Educational Environments

Currently, the only way to gain hands-on experience with industrial control networks is to implement an actual system, but this can be prohibitively expensive. The same is true for features that are specific to physical devices such as uploading and downloading ladder logic programs. Although simulators are available for developing and testing ladder logic programs, it is difficult, if not impossible, to adequately test network communications and functionality without using actual industrial control devices [16, 17].

In an academic setting, emulators can be used to provide a deep understanding of the tools and techniques used to conduct attacks on industrial control

networks. Indeed, an emulator can be used to extend the application of tools such as Nmap [8] and Metasploit [15] to industrial control networks. In particular, attacks on a PLC emulator can help demonstrate traditional information technology attacks as well as their effects on an industrial control system. An emulator that responds to scans and attacks from tools such as Nmap and Metasploit in the same manner as a real PLC can provide hands-on attack experience without putting the industrial control system at risk. Also, an emulator can be remotely setup by instructors to model any number of device configurations and vulnerabilities and record the techniques used by students to exploit the device.

A PLC emulator must have high fidelity; for example, it should accurately mimic the responses of a real PLC to fingerprinting and port scanning tools. However, even a lower-accuracy emulator has value in a training environment – students could learn to identify control system devices by employing network scanning and protocol dissection techniques.

3.3 Advantages of Emulation

Industrial control devices are notoriously sensitive due their lack of input sanitization and use of non-robust communications protocols. Even seemingly benign network mapping tools can cause serious problems. A 2005 Sandia report [3] describes several instances where network scans of SCADA networks disrupted industrial control systems. The reported behavior included erratic movements of a nine-foot robotic arm in an area shared with human operators, a malfunction of a microchip manufacturing process that resulted in $50,000 damage to electrical components, and a large natural gas delivery failure due to unresponsive SCADA equipment.

A significant advantage of an emulator is that it retains the robustness of the underlying system architecture. Even if the emulator locks up – i.e., it becomes "bricked" – there is no flash bootloader to accidentally overwrite or electronically-erasable programmable read-only memory (EEPROM) to accidentally erase because it is implemented purely in software. A simple reset of the software returns the emulator to its normal operating state, unlike a real PLC, which would require the use of a hardware debugging device or even the replacement of PLC components.

An industrial control emulator is cost effective compared with control system hardware, especially when a single PLC can easily cost more than $10,000. Power constraints are also a factor – incorporating large numbers of PLCs in an educational or training environment may not be possible due to power requirements.

An emulator is scalable and is easily ported to different platforms. Multiple emulators can be chained together without serious interoperability problems. Also, emulators can be configured to model different vendor platforms (e.g., Siemens, Allen-Bradley and Koyo) and myriad network parameters (e.g., MAC address, IP address, host name and ID).

4. Emulator Design Considerations

Despite the fact that industrial control systems are fragile, quickly antiquated from a technology standpoint and often designed without security considerations, they are expensive, complex and invariably networked with other devices. Most modern PLCs provide application-level services such as web configuration interfaces, file transfer services, remote programming services and remote monitoring services through a network interface. All these services have to be replicated in an emulator.

The accuracy of an emulator can be assessed using observable characteristics that a user would leverage to establish device authenticity. These attributes may be derived from the standard process that an attacker would use to discover, interrogate, fingerprint and exploit a networked device [11]. In general, four accuracy considerations exist for an emulator:

- **Superficial Accuracy:** This is established when a user believes that a device is authentic based on what is visually observed, such as the appearance of the web interface.
- **Packet-Level Accuracy:** This is established when a user believes that a device is authentic based on the packets that are sent to and received from the device.
- **Scanning Tool Accuracy:** This is established when a user believes that a device is authentic based on the results produced by an automated scanning tool.
- **Attack Tool Accuracy:** This is established when a user believes that a device is authentic based on the success achieved by an exploitation tool.

Like a network emulation, a control system emulation must accurately reflect the operational environment. Accuracy is a measure of the realism of the emulation compared with the actual device. For example, an emulation of a web interface in a modern PLC is accurate if the webpage has the same appearance (e.g., banner, form elements and graphics) and provides the same functionality. Additionally, the packets transmitted during query and response interactions must match the packets transmitted to and received from the actual PLC.

5. Emulator Implementation

This section provides implementation details regarding the emulator along with an evaluation of the accuracy of the emulator.

5.1 Implementation Details

The emulator runs on a Dell Latitude D630 with 2 GB RAM, Intel Core2 Duo CPU 2.00 GHz processor running VMware with Linux Ubuntu 2.6.35. The Linux kernel was built with `iptables` and Netfilter NFQUEUE modules to provide the necessary Linux firewall and user-space network packet functionality.

The emulated PLC comprised the Koyo DirectLogic 405 CPU with a D4-08ND3S digital input module, D4-08TR digital relay module and H4-ECOM-100 Ethernet communications module. This PLC was selected because it demonstrated certain security vulnerabilities, including the vulnerability to an automated method that cracks device passwords [13, 18]. The emulated firmware used in our work was version H4-ECOM 4.0.1735, which contains a known vulnerability and an associated Metasploit module `koyo_login` [18].

The emulator is written in C and includes several interpreted Python modules. The emulator incorporates seven simultaneously running user-space processes in addition to the `iptables` firewall kernel module. It offers three network services to users: (i) a web server on TCP port 80 providing a configuration interface; (ii) a Modbus TCP service on TCP port 502; and (iii) a Host Automation Products (HAP) service on UDP port 28784. The Modbus TCP service is indicative of the Koyo PLC implementation of the common industrial protocol service. The HAP service enables remote management and configuration of the Koyo PLC.

For purposes of this research, the Modbus service was replicated strictly for device fingerprinting associated with the Koyo PLC. Because the research focus was on the Koyo PLC application and not messaging protocol standards, the Modbus functionality was not evaluated. Note, however, that Modbus functionality is readily incorporated as demonstrated by Berman, *et al.* [1].

5.2 Emulator Accuracy

The emulated PLC was evaluated for consistency with the Koyo PLC in terms of superficial accuracy, packet-level accuracy, scanning tool accuracy and attack tool accuracy. Each of the two emulated services, web and HAP, had standard and non-standard types of interactions, yielding a total of four types of queries. These queries were selected for the evaluation because they are representative of the interactions associated with an attacker targeting an Internet-connected PLC.

- **Standard Web Queries:** These queries are associated with interactions with a web server listening on TCP port 80. The queries represent normal, within-bounds requests from an Internet Explorer web browser.
- **Non-Standard Web Queries:** These queries are associated with OS fingerprinting scans using Nmap that attempt to identify a device based on the responses to non-standard TCP, IP and ICMP packets sent to the device. By default, Nmap chooses the lowest open TCP port on a device to perform its tests. Therefore, Nmap selects TCP port 80 unless it is directed otherwise.
- **Standard HAP Queries:** These queries are associated with interactions with the HAP industrial protocol server listening on UDP port 28784. These queries represent normal, within-bounds requests from NetEdit3 [6], a free vendor tool.

No.	Dest Addr	Src Addr	Protocol	Length	Info
2	10.1.0.147	10.1.0.79	TCP	60	http > 5000 [SYN, ACK] Seq=2374868006 Ack=1606421648 Win=2048 Len=0 MSS=512
5	10.1.0.147	10.1.0.79	TCP	60	http > 5000 [ACK] Seq=2374868007 Ack=1606421904 Win=2048 Len=0
6	10.1.0.147	10.1.0.79	HTTP	566	HTTP/1.1 200 OK (text/html)
8	10.1.0.147	10.1.0.79	HTTP	566	Continuation or non-HTTP traffic
10	10.1.0.147	10.1.0.79	HTTP	566	Continuation or non-HTTP traffic
12	10.1.0.147	10.1.0.79	HTTP	494	Continuation or non-HTTP traffic
14	10.1.0.147	10.1.0.79	TCP	60	http > 5000 [FIN, ACK] Seq=2374869983 Ack=1606421904 Win=2048 Len=0
16	10.1.0.147	10.1.0.79	TCP	60	http > 5000 [ACK] Seq=2374869984 Ack=1606421905 Win=2048 Len=0

Emulator response.

No.	Dest Addr	Src Addr	Protocol	Length	Info
2	10.1.0.147	10.1.0.93	TCP	60	http > 5000 [SYN, ACK] Seq=540754409 Ack=3851000542 Win=2048 Len=0 MSS=512
5	10.1.0.147	10.1.0.93	TCP	60	http > 5000 [ACK] Seq=540754410 Ack=3851000798 Win=2048 Len=0
6	10.1.0.147	10.1.0.93	HTTP	566	HTTP/1.1 200 OK (text/html)
8	10.1.0.147	10.1.0.93	HTTP	566	Continuation or non-HTTP traffic
10	10.1.0.147	10.1.0.93	HTTP	566	Continuation or non-HTTP traffic
12	10.1.0.147	10.1.0.93	HTTP	494	Continuation or non-HTTP traffic
14	10.1.0.147	10.1.0.93	TCP	60	http > 5000 [FIN, ACK] Seq=540756386 Ack=3851000798 Win=2048 Len=0
16	10.1.0.147	10.1.0.93	TCP	60	http > 5000 [ACK] Seq=540756387 Ack=3851000799 Win=2047 Len=0

Koyo PLC response.

Figure 1. Web server responses showing differences in TCP congestion window sizes.

- **Non-Standard HAP Queries:** These queries are associated with attacks on HAP protocol security measures using the `koyo_login` Metasploit module to exploit vulnerabilities.

The accuracy of the responses to the four types of queries are discussed below. The results are useful in educational environments where students learn scanning techniques and attack strategies that target industrial control systems. Additionally, the results showcase the applicability of the emulator in control system research efforts.

- **Standard Web Responses:** To analyze the superficial accuracy, the Koyo website HTML source was used to duplicate the original content and visual representation. Although it appears to be trivial, the notion of superficial accuracy should not be underestimated. For example, in a honeypot application, even a slight visual difference can alert a potential attacker that the device is not authentic.

  Web pages were transmitted byte-for-byte from the PLC to the user's browser. As such, packet-level accuracy is an important consideration in PLC emulation. A single web query to `index.html` of the Koyo PLC generates 2,432 bytes. Excluding the non-deterministic header fields (e.g., TCP sequence and acknowledgement numbers), there are 2,330 bytes that are expected to be identical. The emulator differs by only five bytes of the 2,330 total bytes, demonstrating a consistency of 99.8%.

  Four of the five differences occur in the TCP response header as a result of the Koyo PLC TCP/IP stack implementation handling procedures for the TCP push flag and TCP congestion window. Figure 1 shows the web server responses from the emulator and the Koyo PLC for different TCP congestion window sizes. Correcting these differences requires the modification and recompilation of the Linux kernel.

The fifth difference is the result of an implementation error in the emulator that appears in the `index.html` web page. The emulator value is `0x3a` (ASCII ":" character) while the Koyo PLC value is `0x0a` (ASCII line feed character). Because this difference occurs with a whitespace character, it is likely the result of a copy-and-paste error. Whatever the cause, the difference is readily fixed by changing the `index.html` source code of the emulator.

- **Non-Standard Web Responses:** An Nmap operating system scan uses TCP/IP stack fingerprinting to identify the operating system of a target device [10]. From the point of view of scanning accuracy, the emulator must respond to an Nmap operating system scan in a manner that is consistent with the Koyo PLC. Indeed, the scan should provide no indications that the emulator is running a Linux operating system within a virtual environment.

  Figure 2 shows the results obtained for non-standard web queries to the emulator and Koyo PLC. The Nmap scan results differ only in the TCP sequence number prediction fields (SP and ISR) and in the unused ICMP port unreachable field non-zero field (UN). The corresponding scanning tool accuracy is 97.25%. Note that an Nmap scan on a similar VMware configuration with no installed emulator correctly identified the Linux operating system. The results indicate that the emulator sufficiently conforms to the Koyo PLC with regard to the operating system characteristics gleaned using external communications.

- **Standard HAP Responses:** Standard HAP requests were sent using the UDP transmission protocol. In the experiment, 305 of the 309 total bytes in the HAP response from the Koyo PLC were deterministic, and the emulator was 100% accurate for the deterministic bytes. Figure 3 shows the response bytes for HAP protocol queries to the emulator and Koyo PLC. Note that the non-deterministic fields are highlighted and include the Ethernet address, IP header checksum, IP source address and UDP header checksum.

  The accurate responses produced for standard HAP requests indicate the emulator is capable of interacting with any tool in a manner that conforms with the HAP protocol standard. As such, engineering modifications and device configuration updates are appropriately handled by the emulator, consistent with the expected operations of the Koyo PLC.

- **Non-Standard HAP Responses:** The attack tool accuracy for the Metasploit module was determined based on successful attack execution and device exploitation. Figure 4 demonstrates an execution of the `koyo_login` Metasploit module to brute-force the emulator password. The exploit works in the same manner as the Koyo PLC and discovers the correct emulator password ("A0000322"). Thus, the emulator can be

```
Not shown: 131067 closed ports
PORT      STATE         SERVICE
80/tcp    open          http
502/tcp   open          asa-appl-proto
28784/udp open|filtered unknown
MAC Address: 00:E0:62:60:46:24 (Host Engineering)
Device type: specialized
Running: Koyo embedded
OS details: Koyo DirectLogic PLC
TCP/IP fingerprint:
OS:SCAN(V=5.21%D=12/13%OT=80%CT=1%CU=1%PV=Y%DS=1%DC=D%G=Y%M=00E062%TM=50CA4
OS:A87%P=x86_64-unknown-linux-gnu)SEQ(SP=5D%GCD=1%ISR=C5%TI=Z%CI=Z%II=RI%TS
OS:=U)OPS(O1=M200%O2=M200%O3=M200%O4=M200%O5=M200%O6=M109)WIN(W1=800%W2=800
OS:%W3=800%W4=800%W5=800%W6=800)ECN(R=Y%DF=Y%T=FF%W=800%O=M200%CC=N%Q=)T1(R
OS:=Y%DF=Y%T=FF%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=Y%T=FF%W=800%S=O%A=S
OS:+%F=AS%O=M109%RD=0%Q=)T4(R=Y%DF=Y%T=FF%W=800%S=A+%A=S%F=AR%O=%RD=0%Q=)T5
OS:(R=Y%DF=Y%T=FF%W=800%S=A%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=FF%W=800%S=A
OS:%A=S%F=AR%O=%RD=0%Q=)T7(R=Y%DF=Y%T=FF%W=800%S=A%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=Y%T=FF%IPL=38%UN=B390%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DF
OS:I=S%T=FF%CD=S)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=93 (Good luck!)
IP ID Sequence Generation: All zeros
Read data files from: /usr/share/nmap
OS detection performed. Please report any incorrect results at http://
Nmap done: 1 IP address (1 host up) scanned in 3657.08 seconds
           Raw packets sent: 159858 (5.634MB) | Rcvd: 131086 (6.293MB)
```

Nmap OS scan result.

```
Not shown: 131067 closed ports
PORT      STATE         SERVICE
80/tcp    open          http
502/tcp   open          asa-appl-proto
28784/udp open|filtered unknown
MAC Address: 00:E0:62:60:46:23 (Host Engineering)
Device type: specialized
Running: Koyo embedded
OS details: Koyo DirectLogic PLC
TCP/IP fingerprint:
OS:SCAN(V=5.21%D=12/15%OT=80%CT=1%CU=1%PV=Y%DS=1%DC=D%G=Y%M=00E062%TM=50CCF
OS:062%P=x86_64-unknown-linux-gnu)SEQ(SP=2F%GCD=1%ISR=9A%TI=Z%CI=Z%II=RI%TS
OS:=U)OPS(O1=M200%O2=M200%O3=M200%O4=M200%O5=M200%O6=M109)WIN(W1=800%W2=800
OS:%W3=800%W4=800%W5=800%W6=800)ECN(R=Y%DF=Y%T=FF%W=800%O=M200%CC=N%Q=)T1(R
OS:=Y%DF=Y%T=FF%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=Y%T=FF%W=800%S=O%A=S
OS:+%F=AS%O=M109%RD=0%Q=)T4(R=Y%DF=Y%T=FF%W=800%S=A+%A=S%F=AR%O=%RD=0%Q=)T5
OS:(R=Y%DF=Y%T=FF%W=800%S=A%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=FF%W=800%S=A
OS:%A=S%F=AR%O=%RD=0%Q=)T7(R=Y%DF=Y%T=FF%W=800%S=A%A=S+%F=AR%O=%RD=0%Q=)U1(
OS:R=Y%DF=Y%T=FF%IPL=38%UN=D046%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DF
OS:I=S%T=FF%CD=S)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=47 (Good luck!)
IP ID Sequence Generation: All zeros
Read data files from: /usr/share/nmap
OS detection performed. Please report any incorrect results at http://
Nmap done: 1 IP address (1 host up) scanned in 125.85 seconds
           Raw packets sent: 131091 (4.721MB) | Rcvd: 131085 (6.292MB)
```

Koyo PLC scan result.

Figure 2. Nmap operating system scan results.

```
0000   00 1c 23 1a 37 21 00 e0   62 60 46 25 08 00 45 00   ..#.7!..b`F%..E.
0010   01 27 00 00 40 00 ff 11   65 e2 0a 01 00 4f 0a 01   .'..@...e....O..
0020   00 93 70 70 13 88 01 13   e5 80 48 41 50 25 00 bc   ..pp......HAP%..
0030   c3 02 01 00 00 48 34 2d   45 43 4f 4d 31 30 30 20   .....H4-ECOM100
0040   45 74 68 65 72 6e 65 74   20 43 6f 6d 6d 75 6e 69   Ethernet Communi
0050   63 61 74 69 6f 6e 73 20   4d 6f 64 75 6c 65 2e 00   cations Module..
0060   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
0070   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
```

Emulator response.

```
0000   00 1c 23 1a 37 21 00 e0   62 60 46 23 08 00 45 00   ..#.7!..b`F#..E.
0010   01 27 00 00 40 00 ff 11   65 dd 0a 01 00 54 0a 01   .'..@...e....T..
0020   00 93 70 70 13 88 01 13   e5 7b 48 41 50 25 00 bc   ..pp.....{HAP%..
0030   c3 02 01 00 00 48 34 2d   45 43 4f 4d 31 30 30 20   .....H4-ECOM100
0040   45 74 68 65 72 6e 65 74   20 43 6f 6d 6d 75 6e 69   Ethernet Communi
0050   63 61 74 69 6f 6e 73 20   4d 6f 64 75 6c 65 2e 00   cations Module..
0060   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
0070   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
```

Koyo PLC response.

Figure 3. Standard HAP packet responses.

```
Module options (auxiliary/scanner/scada/koyo_login):

   Name              Current Setting       Required       Description
   ----              ---------------       --------       -----------
   PREFIX            A                     yes            The prefix to use for the password (default: A)
   RECV_TIMEOUT      3                     no             Time (in seconds) to wait between packets
   RHOSTS            10.1.0.79             yes            The target address range or CIDR identifier
   RPORT             28784                 yes            The target port
   THREADS           1                     yes            The number of concurrent threads

msf  auxiliary(koyo_login) > exploit

[*] 10.1.0.79:28784 - KOYO - Checking the controller for locked memory...
[*] 10.1.0.79:28784 - KOYO - Controller locked; commencing bruteforce...
[+] 10.1.0.79:28784 - KOYO - Found passcode: A0000322...
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf  auxiliary(koyo_login) >
```

Figure 4. Metasploit module `koyo_login` executing successfully on the emulator.

used quite effectively in research and educational settings to demonstrate PLC exploitation techniques.

Obviously, it is important that the emulator behave in a manner that is consistent with the Koyo PLC. Table 1 shows the emulator results for the various types of accuracy associated with the two emulated services. When emulating the standard web and standard HAP services, the emulator is accurate at the superficial and packet levels. The non-standard web service is accurate at the scanning level – when a user scans the emulator using Nmap, the scanning results confirm the authenticity of the device. Finally, the non-standard HAP service is accurate at the attack tool level because the emulator is vulnerable to a Metasploit attack consistent with the Koyo PLC.

Table 1. Emulator services and accuracy considerations.

Service	Query Type	Superficial	Packet	Scan	Attack
Web	Standard (Browser)	X	X		
	Non-Standard (Nmap)			X	
HAP	Standard (NetEdit3)	X	X		
	Non-Standard (Metasploit)				X

6. Conclusions

The PLC emulator implemented in the VMware virtual environment provides a cost-effective and scalable emulation solution. Experimental evaluations demonstrate that the emulator accurately replicates a real PLC with respect to standard and non-standard web and HAP services.

We hope that this work stimulates renewed efforts at developing sophisticated industrial control system emulators for research and educational environments. Industrial control system emulators provide significant advantages over hardware-based testbeds and simulation environments. Their robustness, versatility and configurability are attractive for understanding, developing and analyzing defensive and offensive techniques, more so because these efforts can be conducted without concern for damaging costly equipment.

Note that the views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Air Force, U.S. Department of Defense or the U.S. Government.

References

[1] D. Berman and J. Butts, Towards a characterization of cyber attacks on industrial control systems: Emulating field devices using Gumstix technology, *Proceedings of the Fifth International Symposium on Resilient Control Systems*, pp. 63–68, 2012.

[2] C. Davis, J. Tate, H. Okhravi, C. Grier, T. Overbye and D. Nicol, SCADA cyber security testbed development, *Proceedings of the Thirty-Eighth North American Power Symposium*, pp. 483–488, 2006.

[3] D. Duggan, Penetration Testing of Industrial Control Systems, Sandia Report SAND2005-2846P, Sandia National Laboratories, Albuquerque, New Mexico, 2005.

[4] Flux Research Group, Network Emulation Testbed, School of Computing, University of Utah, Salt Lake City, Utah (`www.emulab.net`).

[5] K. Gatens, INEEL establishes National SCADA Test Bed, Feature Story, Idaho National Laboratory, Idaho Falls, Idaho (`www.inl.gov/featurestories/2003-03-25.shtml`), 2003.

[6] Host Engineering, NetEdit v3, Jonesborough, Tennessee (`www.hosteng.com/SW-Products/NetEdit3.htm`).

[7] Idaho National Engineering and Environmental Laboratory, INEEL establishes National SCADA Testbed, *Need to Know*, vol. 3(2), pp. 1–3, 2003.

[8] Insecure.com, Nmap Security Scanner, Sunnyvale, California (`nmap.org`).

[9] N. Kisserli, D. Schellekens and B. Preneel, Self-encrypting code to protect against analysis and tampering, *Proceedings of the First Benelux Workshop on Information and System Security*, 2006.

[10] G. Lyon, Nmap Network Scanning, The Official Nmap Project Guide to Network Discovery and Security Scanning, Insecure.com, Sunnyvale, California, 2008.

[11] S. McClure, J. Scambray and G Kurtz, *Hacking Exposed 7: Network Security Secrets and Solutions*, McGraw-Hill, Emeryville, California, 2012.

[12] T. Morris, R. Vaughn and Y. Dandass, A testbed for SCADA control system cyber security research and pedagogy, *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research*, article no. 27, 2011.

[13] D. Peck and D. Peterson, Leveraging Ethernet card vulnerabilities in field devices, *Proceedings of the SCADA Security Scientific Symposium*, 2009.

[14] C. Queiroz, A. Mahmood, J. Hu, Z. Tari and X. Yu, Building a SCADA security testbed, *Proceedings of the Third International Conference on Network and System Security*, pp. 357–364, 2009.

[15] Rapid7, Metasploit Framework, Boston, Massachusetts (`www.metasploit.org`).

[16] Rockwell Automation, GuardPLC Safety Control Programming Software, Milwaukee, Wisconsin.

[17] Siemens, LOGO! Software, Munich, Germany.

[18] R. Wightman, Koyo/Automation direct vulnerabilities, Digital Bond, Sunrise, Florida (`www.digitalbond.com/blog/2012/02/08/koyoautomation-direct-vulnerabilities`), 2012.