Global Information Assurance Certification Paper

Forensic Analysis of Industrial Control Systems

GIAC (GCFA) Gold Certification

Author: Lew Folkerth, lew.folkerth@rfirst.org

Advisor: Richard Carbone

Accepted: September 24, 2015

Abstract

Industrial Control Systems (ICS) contribute to our safety and convenience every day, yet remain unseen and unnoticed. From oil refineries to traffic lights, from the elevators we ride to the electric power plants that keep our lights on, they provide the control and monitoring for our essential services. ICS have served reliably for decades, but a changing technological environment is exposing them to risks they were not designed to handle. Internet connectivity, vulnerability assessment tools, and attacks by criminal and nation-state organizations are part of this changing picture. Along with this higher-risk environment comes the certainty that some of our ICS will be compromised. In order to prevent recurring attacks, security professionals must be able to discover where the compromise originated, how it was carried out, and, if possible, who was responsible. Many types of ICS run on proprietary hardware, so commonly accepted forensic techniques must be adapted for use in an ICS environment. In order to detect a compromise, baseline configurations should be documented. Networks should be monitored for unauthorized access and activity. In addition, a response plan should be in place to maintain service and streamline recovery. Techniques for forensic analysis were adapted and tested on live ICS, resulting in recommendations for successful detection and recovery after an incident. With adequate preparation and the appropriate response planning and execution, it is possible to successfully perform a forensic analysis for an ICS compromise.

1. Introduction

1.1. ICS Architecture

Industrial Control Systems (ICS) keep our world running. Applications for ICS include electric power generation, transmission, and distribution, refinery control, and manufacturing automation. ICS can be loosely grouped into three types, based on the architecture of the system:

- Supervisory Control and Data Acquisition (SCADA) – These systems use a central computer that communicates to multiple Remote Telemetry Unit (RTU) "field" devices. Data is collected from the RTUs and processed at the central computer. Control of remote functions via the RTUs is also initiated from the central computer, hence "Supervisory" in the name. The central computer provides Human-Machine Interface (HMI) and data historian functions (Stouffer, Pillitteri, Lightman, Abrams, & Hahn, 2015, p. 2.5). Utilities engaged in transmission and distribution of electricity, natural gas, and water make extensive use of SCADA, as do many other industries (Lee, 2013).
- Digital Control System (DCS) – Where a SCADA system centralizes control logic in a central platform, a DCS distributes the control logic closer to the processes being controlled. The field level devices providing the control logic are then managed by supervisory level systems. Similar to SCADA, the supervisory level systems provide HMI and data historian functions. Chemical manufacturing plants and electric power plants are typical uses for a DCS. (Stouffer, Pillitteri, Lightman, Abrams, & Hahn, 2015, p. 2.12)
- Non-centralized systems – Some control systems do not need central management. In such cases, a programmable logic controller or other control device may be configured as a combination control system, HMI, and data historian. This configuration may typically be found in manufacturing processes, among others. (Stouffer, Pillitteri, Lightman, Abrams, & Hahn, 2015, p. 2.12)

1.2. ICS Components

The control system architectures above are made up of individual components that together provide the necessary functionality. These components may include:

- General-purpose computers – Computers such as laptops, desktops, workstations, and servers which run their applications on top of a common operating system such as Windows, Unix, or Linux. These computers may be engaged directly in the control process, but more often are found providing data historian, programming, and supervisory functions.
- Programmable Logic Controller (PLC) – The PLC is the building block of many automation systems. A PLC is typically designed as a general control system with many options for input and output. PLCs may be used as a node in a centralized system under the control of a SCADA or DCS server. They may also function as stand-alone systems or participate in a network of PLCs. PLCs are commonly programmed using a technique known as "ladder logic." In a ladder logic program, functionality is based on "rungs," with multiple rungs providing different functions. Ladder logic rungs resemble the rungs on a ladder, hence the name. This is done to emulate the hardware control logic that PLCs were developed to replace. Instead of having a hardware rack with rails of relays, timers, etc., the control system engineer replicates the hardware in a ladder logic program. (Fehr, 2003)
- Remote Telemetry (or Terminal) Units (RTUs) – An RTU may be thought of as a communication hub located near the devices it communicates with. A typical component of a SCADA system, the RTU is usually used as both a remote hub for collecting data from field devices, and as a relay point for control commands.
- Special purpose systems – Other types of ICS components may be designed for a specific task. For example, an HMI panel may be used as the human interface to a PLC.
- Smart sensors and actuators – Control systems obtain data from sensors and provide control functions via actuators. So-called "smart" sensors and actuators

are able to communicate with standard protocols such as Fieldbus (Stouffer, et al., 2015, p. 5-19). Some of these protocols may also ride on IP, opening the possibility of access to these sensors and actuators by unauthorized actors. (ODVA, 2008)

1.3. ICS Environment

The security practitioner should be aware that control systems operate in modes and environments not typically faced by an IT professional:

- Control systems frequently operate in harsh environments such as factory floors or electrical substation control houses. This limits the type of devices that may be deployed. Control systems must function in extremes of temperature and humidity, and many systems are subjected to vibration and contaminants that are foreign to a data center. Workstations, servers, and peripherals that operate well in an office or data center environment may not survive in the environment required by a control system.
- Control systems operate in real-time, where response times and network latency must be strictly controlled. This may limit deployment of traditional IT security tools. For example, it may be desirable to segment a plant network to isolate smart sensors and actuators for the rest of the control system network. But placing a firewall in such a network may increase network latency to an unacceptable level, causing processes to fail.
- Control systems have unusually high uptime requirements compared to IT systems. A patch needed to secure a DCS running a power plant may need to be placed on hold until the plant has a window of scheduled downtime. Such a window could take months to arrive, necessitating an alternate method of protecting the DCS in the interim.
- Some of the proprietary or legacy protocols in use may not be supported by typical IT security tools such as firewalls or intrusion detection systems (IDS) (Horkan, 2015, pp. 23-24). For example, Foundation Fieldbus HSE makes use

of IP multicast features (Stouffer, Pillitteri, Lightman, Abrams, & Hahn, 2015, p. 5.20), which may not be well supported by some security technologies.

- Control systems have life expectancies that are longer than typical IT products. One side effect of this disparity is that control systems may continue to operate using platforms or versions that are no longer supported by vendors. Examples of this include Vax computers and versions of Windows NT and XP.

Some of the operating conditions experienced by a typical control system may work to the benefit of the security practitioner:

- Control system network configurations are usually more stable than their IT counterparts. While IT networks may change frequently based on equipment being added and removed, addition or removal of a device on a typical control system network is rare. This may assist in spotting unauthorized devices on the control system network.
- Traffic patterns on the control system network should be relatively static. This may assist in the configuration of an IDS or other monitoring system.
- Many control system components use a non-routable serial protocol for communications. While this makes monitoring such traffic very difficult, it also reduces the risk of compromise, and changes the threat environment to that of an insider only.

1.4. Need for ICS Forensic Capability

In the past, if a control system failed to function properly, the first priority of the control system engineer was to return the system to operational status. Today, however, forensic considerations must be acknowledged in order to ensure that the control system was not compromised by a hostile actor. If a hostile act occurred, then action must be taken to identify the compromise and harden the control system against similar attacks. Regulatory agencies may require reporting of incidents. The accepted six-step Incident Response Process: Preparation, Identification, Containment, Eradication, Recovery, and

Lessons Learned (Kral, 2011) may need to be modified, and the associated forensic techniques will need to be adapted to the control system environment as well.

One common mistake in developing forensic techniques for ICS is to concentrate on the central server of a DCS or SCADA system. One reason for this is that these servers are usually built upon a well-understood operating system, and thus incident responders have the potential of using freely available forensic tools for these platforms. However, it is important to also consider field devices like a PLC or RTU in a forensics program. Few, if any, common tools are available for these platforms. This paper will examine some of the options available in detecting and responding to compromises of these platforms.

1.5. Prior Work

Many works are available that discuss ICS security, such as *Robust Control System Networks* (Langner, 2012), *Guide to Industrial Control Systems (ICS) Security* (Stouffer, Pillitteri, Lightman, Abrams, & Hahn, 2015), and *Cybersecurity for Industrial Control Systems: SCADA, DCS, PLC, HMI, and SIS* (Macaulay & Singer, 2011) but little is available that deals specifically with forensic techniques for ICS. Forensic information that is available provides useful techniques for the central SCADA or DCS servers, but only discusses field devices such as PLCs and RTUs at a general level.

In 2008, the Department of Homeland Security's Control Systems Security Program published *Recommended Practice: Creating Cyber Forensics Plans for Control Systems* (Fabro & Cornelius, 2008), which laid the groundwork for future detailed forensic analysis of control systems, spelling out the challenges this field faces.

Who's Really Attacking Your ICS Equipment? (Wilhoit, 2013) discussed the findings of research using a honeypot to determine sources and methods of attack on an ICS. The paper concluded by providing a bullet list of recommended actions to increase ICS security.

The seventh chapter of *Handbook of SCADA/Control Systems Security* discusses forensics management (Wright, 2013). The discussion focuses primarily on the SCADA system, with general mention of some techniques for PLCs and RTUs.

Towards a SCADA Forensics Architecture lays out a framework for applying IT forensic techniques to SCADA servers and acknowledges the difficulty of obtaining tools for traditional forensic analysis of control systems (Wu, Pagna Disso, Jones, & Campos, 2013).

Kim Zetter describes the intensive forensic efforts to decode Stuxnet, a malware platform that infected Windows PCs in order to compromise specific PLCs, in *Countdown to Zero Day: Stuxnet and the World's First Digital Weapon* (Zetter, 2014).

1.6. Scope of Research

For forensic purposes, attacks on a control system may be lumped into three general categories: attacks through the configuration workstation, attacks through the Ethernet network, and non-Ethernet attacks.

Attacks through the configuration workstation may be addressed by performing forensics on the workstation, using the accepted techniques for the operating system of the workstation.

Non-Ethernet attacks will be extremely difficult for the forensic examiner to detect without specialized equipment. For example, Fieldbus uses Manchester coding (Fieldbus Foundation, 2014) which is incompatible with RS-232/RS-488 and Ethernet. For this reason, such attacks are beyond the scope of this paper.

Some common types of attack will not work directly against a field device such as a PLC or RTU. Few control system components, other than those based on common operating systems, are capable of reading removable media such as CDs, DVDs, or USB drives, so these forms of attack should be rare in this environment. Also, phishing and other social engineering attacks may have little success, as these attacks are typically associated with email or web browsing. While this may result in compromise of a SCADA central server or a PLC configuration workstation, direct compromise of a PLC or RTU through these means is unlikely, as PLCs and RTUs are not typically configured with email clients or web browsers.

Historically, field devices such as PLCs have not been designed with security in mind (Peterson, 2013). If a PLC is accessible through a routable network, it may be at

great risk of compromise. Attacks through an Ethernet network, however, may be detectable and forensics performed with readily available tools.

2. ICS Forensics Approach

2.1. Preparation

As with forensic processes in general, preparation for forensic examination of control systems is key.

2.1.1. Control System Baseline

In order to perform the forensic examination of a control system, an accurate baseline of the existing system will be necessary. That baseline should include:

- Hardware configuration of the control system, including schematics or wiring diagrams. The make, model, and serial number should be recorded for each active device in the system. PLC configurations, including any expansion modules, should be documented.
- For PLCs, a copy of all current ladder logic programs should be kept. A previous known-good set of programs should also be available.
- For each device attached to a network, the network address, network settings such as subnet mask and default gateway, MAC address, and listening ports should be documented.
- For each device, documentation of its internal status such as the normal run state of a PLC, trimmer settings, any force settings applied, etc., should be available.

It is important to keep the baseline current. This will be accomplished most effectively by a rigorous change management program. The parts of an effective change management program include documentation, testing, planning, authorization, reversibility, and auditing (Langner, 2012, pp. 151-153).

2.1.2. Network Monitoring

For many types of ICS, such as PLCs, performing forensics on the device itself may not yield the desired results. Proprietary architectures and protocols are often used, and forensic tools are not widely available for these platforms. In this case, the security practitioner must rely on other sources of information. One such source is the control system network.

IT network monitoring techniques and tools may have limitations and must be adapted to the control system environment. For example, configuring an intrusion detection system (IDS) may entail significant difficulties. Protocols such as CIP (Common Industrial Protocol) and EtherNet/IP (CIP over Ethernet) do not appear to be well understood by IDS providers at this time. (Horkan, 2015, pp. 17-19).

For control system networks employing Ethernet communications, it is possible to set up monitoring nodes for the capture of network traffic to significant control system devices such as PLCs. These monitoring nodes may be attached to the control system network with the addition of a port-mirroring switch or network tap. The monitoring node then listens to the mirror port and does not transmit packets on the control system network. A packet capture utility runs on the monitoring node and saves packets to non-volatile memory. The packet capture utility is configured to only save packets of interest. This saves space in non-volatile memory and aids analysis by removing expected traffic from the analysis phase. The non-volatile memory may be removed at any time and the captured packets analyzed for hostile or unexpected behavior.

2.1.3. Logging

If a device is capable of logging events, then logging should be enabled to the extent possible without disrupting the control system.

2.1.4. Tools

In preparing for incident response and forensic analysis, it is important to have the proper tools ready. In addition to the tools normally used for incident response on an IT network, a control system “jump kit” may include:

- A control system configuration workstation, preferably portable, containing the programming software for each type of PLC and smart device that could be encountered. This system should be kept patched and updated.
- An isolated environment for PLC examination. This environment should include the tools necessary to configure and test the PLC under controlled conditions.
- Memory capture tools may be available for some platforms (Wu, Pagna Disso, Jones, & Campos, 2013, p. 18). If such tools are available for any of the platforms in use, these tools should be available to the incident response team.
- Documentation of the architectures and protocols in use.
- Evidence bags large enough to handle PLCs and other evidence to be preserved.
- Other forensic tools appropriate for the platforms in use.
- Spares for each type of PLC and smart device that could be encountered.
- Schematics and wiring diagrams.
- Backup of at least the last two known good PLC ladder programs.
- Test procedures to ensure proper operation of the control system before it is returned to service.
- A set of manuals for each type of control system device on the network.
- Personal protective equipment appropriate to the environment, such as: hard hats; safety glasses; hearing protection; steel or composite toe shoes or boots; dielectric sole shoes or boots; flame or flash resistant clothing; non-synthetic fiber clothing.

2.1.5. Incident Response Team

The composition of control system incident response teams will need careful consideration. Skills required by the team include knowledge of the process being controlled, control system engineering, IT incident response, and forensics. For each of

these areas, at least one member of the team should have in-depth knowledge. Also, each member of the team should have at least a basic understanding of all the skills required. For example, a control systems engineer should have a basic knowledge of the process being controlled, incident response, and forensics, while also possessing an expert-level understanding of control systems.

Such a combination of skills must be developed. Initial training will be needed for each of the skills a team member does not possess. Continuing education and periodic exercises will help to keep the skills fresh.

An incident response team must be familiar with all safety procedures for the working environment. This must be the team's first consideration. Training for each team member must include the safety requirements of the control system environment, such as personal protective equipment requirements, and any operational procedures specific to the environment. For example, understanding of lockout/tagout procedures (OSHA, 2002) may be required to ensure that electrical circuits are only energized when it is safe to do so.

2.1.6. Incident Response Plan

A comprehensive incident response plan will be vital to the successful incident response in a control system environment. In addition to the considerations of a typical IT plan, a control system incident response plan should consider:

- Instructions and plans to keep all personnel safe.
- When and how to fulfill mandatory reporting requirements, if applicable.
- Applicable legal and regulatory requirements.
- The approach to be used to address the incident. For example, will restoration be the highest priority, or will evidence gathering for forensic analysis take precedence? This must be decided before an incident so the incident response team does not waste time or destroy evidence that will be needed in later phases of the response.

- The communications section of the incident response plan should include a plan for keeping management and other essential parties, such as control centers, operations staff, regulatory agencies, or law enforcement, informed of the progress of the response.

2.1.7. Training

The control system incident response team must be properly trained in order to be effective. In addition to training in general incident response and forensic skills, the team should receive training on any systems that may be encountered on the control system network. Obsolete versions of operating systems, applications, or device drivers may still be in use, as well as unusual operating system configurations. Any proprietary systems or equipment in use may require specialized training.

Periodic refresher training for the incident response team should be coordinated with incident response exercises to keep the team's skills fresh.

Operations personnel should also receive periodic training. These personnel require the ability to recognize a possible cyber incident and must know how to initiate a response by the proper incident response team. They must also understand that their role is to recognize and report, not to take action unless in response to safety issues or to prevent equipment damage.

2.2. Response

The following sections present possible items in a control system incident response plan. They are presented in checklist format for possible expansion of an established incident response plan to include control system considerations.

The steps in each checklist should be pre-determined as much as possible by the incident response team, and exercised in a controlled environment. When a live incident occurs, the incident response team will then be prepared to respond, collect and analyze evidence, and recover the systems efficiently.

2.2.1. Incident Response Approach

- Safety:

  Are there safety concerns that must be addressed before or during incident response? Is a safety briefing needed? Is protective clothing required? (OSHA, 2003)

- Evidence Collection and Preservation:

  Is evidence to be collected and preserved for forensic analysis (Henry, 2009)? If so, how much delay in recovery is permissible in order to obtain and preserve the evidence in a forensically sound manner? (Witter, 2001)

- Incident Reporting:

  Is the incident required to be reported to a regulatory or law enforcement agency? For example, the electric industry is required to report certain electric disturbances to the Department of Energy (Department of Energy, 2012). If so, who is responsible for the report? When must such a report be made?

- Recovery:

  What are the goals of the recovery process? How much time is permissible for recovery? Is there a specific state to which the process must be returned in order for recovery to be complete (Young, 2013)?

2.2.2. Volatile Evidence Preservation

- Record Control System Status Information:

  Record the state of any displays (such as an LCD) or status lights (color, on, off, or flashing). Photographs or videos should be taken if feasible. The purpose of this is to capture the exact state of the control system before other actions change the state of the system. Note that in photographing a piece of equipment to capture its status lights, a slow shutter speed such as $1/10^{th}$ of a second should be used. LEDs are frequently pulsed too fast for the human eye to detect, but a fast shutter speed may record an LED as off when it is really on.

- Device Memory:

  Obtain as much information about the memory of affected devices as possible. This will probably require special tools and knowledge of how to use them. PLC configuration workstations should have the capability of capturing the "data files" (in some PLCs these are actually part of system memory) associated with the PLC. Ladder logic programs may also be uploaded to the workstation and preserved (Babcock, 2009). Note that "upload" and "download" typically reference data transfer from the PLC's perspective; "upload" is transfer from the PLC, "download" is to the PLC.

- Open Network Connections:

  If an affected device is networked, obtain as much information as possible about any open network connections. Information such as IP addresses, port numbers, etc., should be retained if available. See Example 1 below for information on obtaining open port information using SNMP.

- Time Clock:

  Ideally, time clocks among the various control system devices will be synchronized (Fabro & Cornelius, 2008, pp. 26-27). If this is not the case, note the difference between the time kept by an affected device and actual time as precisely as possible. This will be useful in creating a timeline and in interpreting evidence gathered from the affected device in relation to evidence from other sources.

- Capture Running Programs:

  If working with a PLC or other similar device, upload the running program (e.g., ladder logic) before the device is powered off. This will require use of a configuration workstation or other special purpose tool.

- Firmware:

  Capture or verify an affected device's firmware, if possible. Some tools such as a "flasher" may be available to capture an image of the firmware (Young,

2013, pp. 151-152). Other tools or mechanisms such as JTAG access may work on control systems (XJTAG, n.d.). Any such tool must be tested and the incident response team must be familiar with its operation.

2.2.3. Non-volatile Evidence Preservation

- Network Monitoring:

  Obtain and preserve all packet captures from installed network monitors. If an IDS is used, preserve all applicable evidence from the IDS (Henry, 2009).

- Preserve Affected Devices:

  If spares are available, pull affected devices from service and replace them with properly configured spares. Preserve the original devices as evidence, which means establishing a chain of custody (Witter, 2001).

2.2.4. Evidence Analysis

Since each control system and each intrusion will be different, the items listed below are suggested questions to ask to guide the forensic analysis in its early phases. Remember that to preserve evidence for future use, it is important to work from copies of the evidence.

- What devices were affected?
- What kind of incident is this? Equipment failure; program bug; improper operation of equipment; deliberate attack?
- If a deliberate attack, what kind of attack was it? Denial of service; data exfiltration; attempt to compromise; actual compromise?
- What techniques were used in the compromise?
- What persistence mechanisms were used?
- Can the attack be re-created?
- Can a similar attack be prevented?

- Can an IDS signature be developed to detect a reoccurrence?
- Are any vulnerabilities identified that can be patched or otherwise mitigated?

2.2.5. Restoration Considerations

Restoration of control systems may need special considerations, such as:

- What is needed to safely restart the process?
- Will the network or other support systems need to run in a restricted operational status until forensic analysis is complete?
- If the process is restarted, can a recurrence of the incident be prevented?
- Will the deployment of additional network monitoring nodes be advisable?
- Should IDS logs and packet captures be reviewed more frequently?
- Should IDS logs be directed to a SIEM for monitoring and alerting?
- Should steps for more active monitoring be undertaken?
- Should the IDS signatures be updated?
- Will the restarted process require a heightened level of vigilance? If so, how long should this heightened vigilance be maintained?

2.2.6. Lessons Learned

During the Lessons Learned phase of the recovery process, control systems' concerns may include:

- Were any previously unknown vulnerabilities discovered in the analysis of the attack? If so, these should be reported to ICS-CERT and the applicable vendor.
- How can the control system be hardened against similar attacks?
- What can the IR team do to improve its control system forensic capabilities?
- Are any additional forensic tools needed for future incidents?

3. Forensic Analysis Examples

A small laboratory, shown in Figure 1, was set up using a Cybati Industrial Kit, an additional laptop, and a Raspberry Pi to recreate an ICS for the following examples.

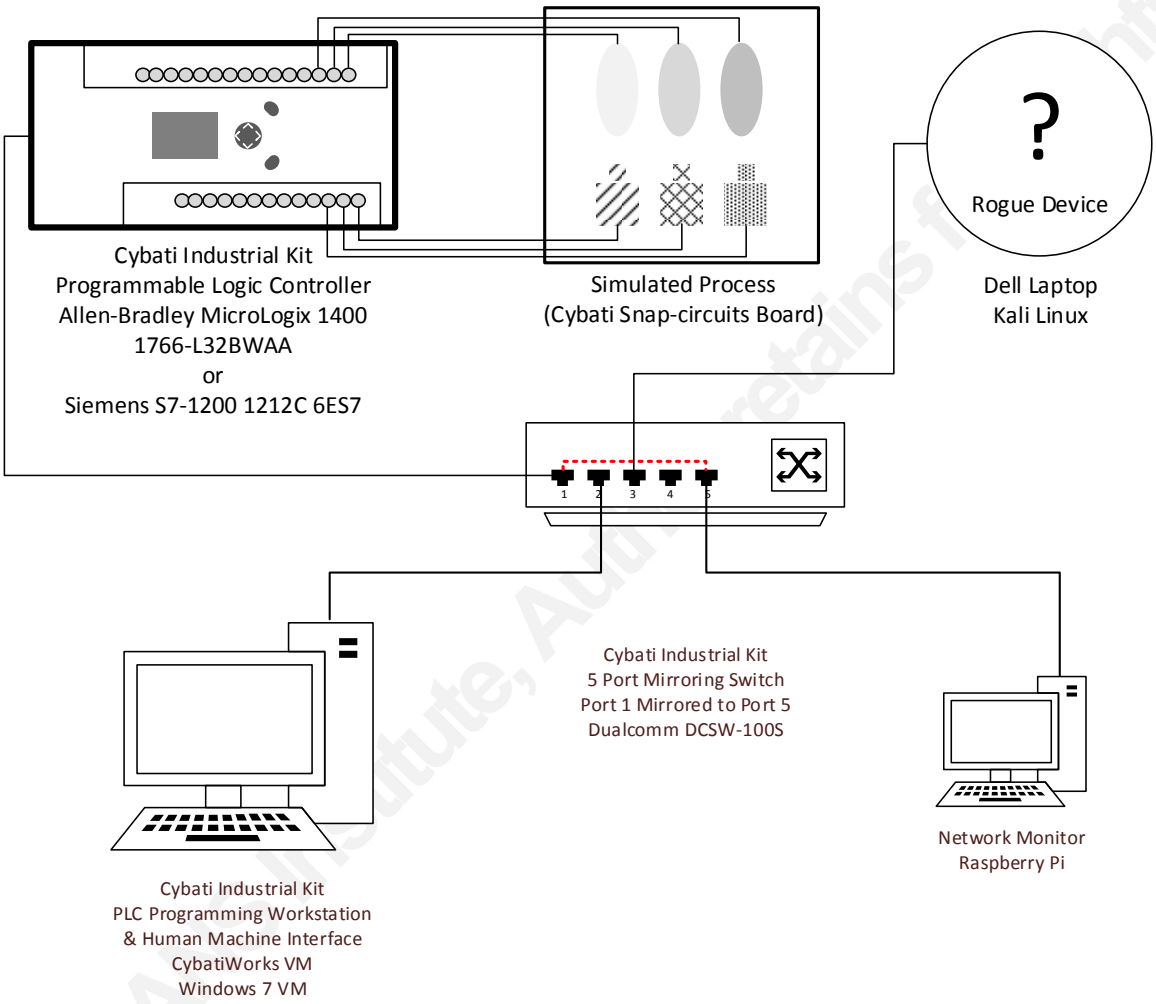


Figure 1: Test Laboratory Configuration

In these examples a network monitor (the Raspberry Pi) running a packet capture utility (tcpdump) is used to keep a history of any unexpected network traffic flowing to the control system. This monitor is attached to the control system network with a port mirroring switch, so that traffic may only flow to the monitor, not from it. Captured packets are stored on the Raspberry Pi's SD card. Packets captured in PCAP format have the advantage of potentially being useful in a timeline analysis when combined with other sources of data (Fletcher, 2015).

The PLC is either an Allen-Bradley MicroLogix 1400 or a Siemens S7-1200, depending on the example. The PLC controls a process simulated using a Cybati-configured Snap Circuits kit.

The "Rogue Device" is a laptop running Kali Linux.

3.1. Example 1 – Reconnaissance

3.1.1. Scenario

A routine review of packet captures from a network monitor at a PLC uncovers access to the PLC by an unexpected device.

3.1.2. Forensic Analysis

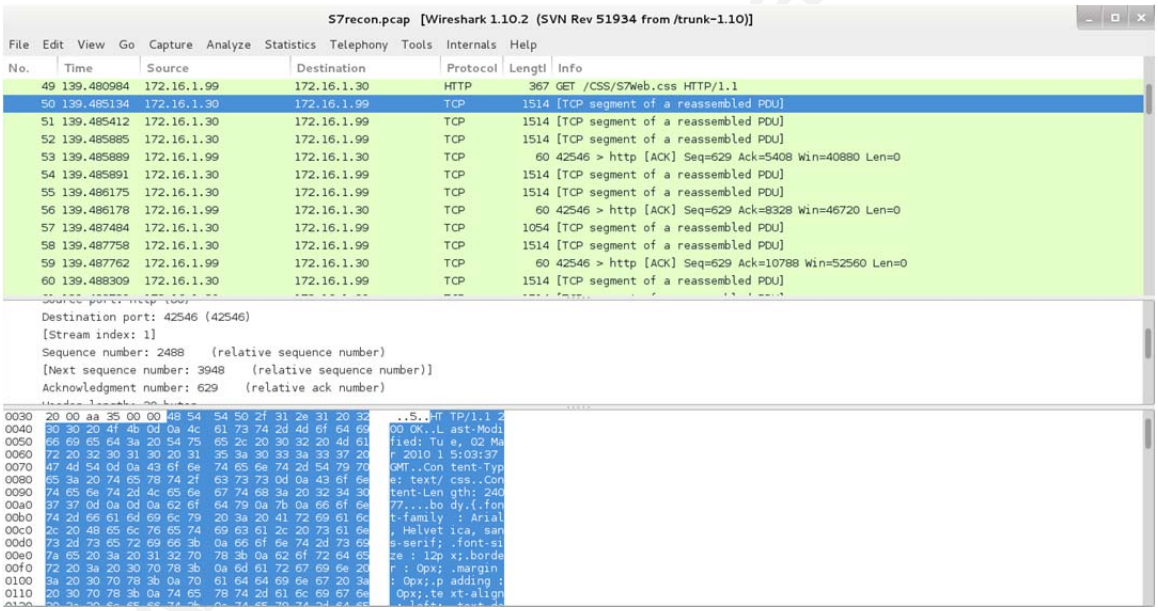


Figure 2: ICS Network Reconnaissance Packet Capture

Lew Folkerth, lew.folkerth@rfirst.org

Figure 2 shows a *Wireshark* display of the network monitor packet capture after network reconnaissance by a rogue network device. The PLC, in this example a Siemens

```
SNMPv2-MIB::sysDescr.0 = STRING: Siemens, SIMATIC S7, CPU-1200,
6ES7 212-1BE31-0XB0, HW: 1, FW: V.3.0.2, SZVCDYEB010206
                                        …
IP-MIB::ipNetToMediaPhysAddress.1.172.16.1.99 = STRING:
5c:26:a:4d:23:83
…
TCP-MIB::tcpConnRemAddress.0.0.0.0.80 = IpAddress: 172.16.1.99
```

Figure 3: Extract of snmpwalk Results

S7, at 172.16.1.30 is displaying its internal web pages to 172.16.1.99, an unauthorized device on the control system network. This type of reconnaissance can reveal sensitive information about the PLC.

Figure 3 shows an extract of the PLC's Management Information Base (MIB) obtained by running *snmpwalk* on the configuration workstation. The information shows the rogue device's MAC address and that it was accessing port 80 on the PLC. It also shows the PLC type, serial and model numbers, and firmware revision. Note that if a rogue device is permitted SNMP access to the PLC, this information may be exfiltrated too.

3.1.3. Lessons Learned

A rogue device should not be permitted access to a control system network. Discovery of such a device almost certainly prevented a compromise of the S7 PLC. This discovery was made possible by the presence of a network monitor and periodic review of its packet captures. Such monitoring nodes in the past have been expensive and difficult to deploy. However, recent gains in technology make devices such as the Raspberry Pi useful in this regard. The Raspberry Pi and the port mirroring switch used in this example are not hardened for control system environments, and therefore may not be useful in monitoring actual control systems. Further research may be able to develop hardened network monitors from present off-the-shelf components.

The review of open connections in the PLC provided additional details about the attacker. Since the connection appears in the SNMP information, it may still be active. If so, it can be traced down and the source of this access identified.

Lew Folkerth, lew.folkerth@rfirst.org

3.2. Example 2 – Denial of Service

3.2.1. Scenario

A manufacturing plant process shut down unexpectedly. During the troubleshooting of the shutdown, an Allen Bradley MicroLogix 1400 located on the plant floor was found in a faulted state. Since the shutdown involved a control system device, the control system incident response team was activated.

3.2.2. Forensic Analysis

Through analysis of captured network traffic, the incident response team found that an Allen-Bradley MicroLogix 1400 PLC was attacked with a denial of service. Figure 4 shows the normal run state of the controller. The power and run lights are on, and the fault light is off. Output bit 4 is on. This bit could control a motor or other element of the process.

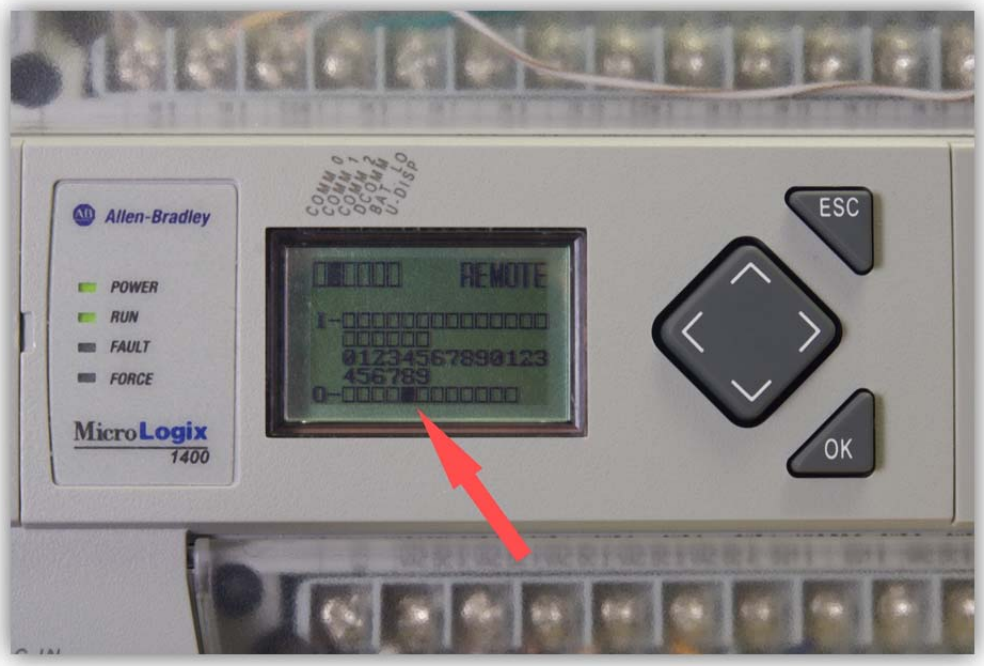
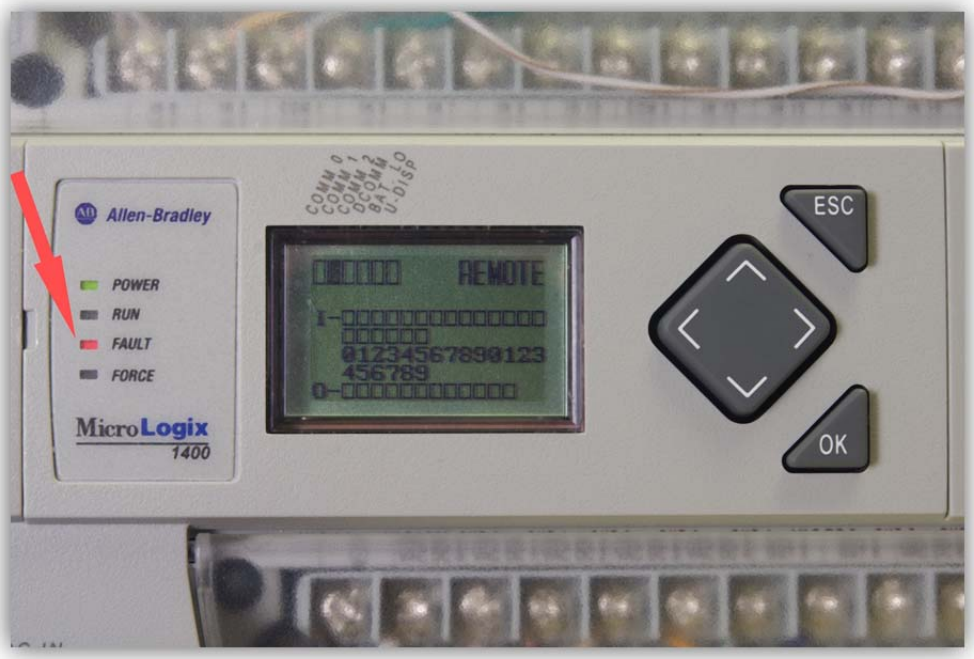


Figure 4: PLC in Pre-faulted State - Note That Output Bit 4 is On (Red Arrow)

After the attack, the controller is in the state shown by Figure 5. Note that the fault light is on, the run light is off, and all output bits are off. For the MicroLogix 1400,

all output bits are set to off when the controller enters a fault state. (Rockwell



Automation, 2014, p. 509)

Figure 5: PLC in Faulted State - Note the Fault Light (Red Arrow)

In the forensic analysis of the incident, the network monitor (Figure 6) revealed that a rogue device (172.16.1.99) sent an EtherNet/IP packet stream to the PLC (172.16.1.30). A reading of the packet stream, which was not decoded in detail by *Wireshark*, would reveal that the PLC was forced to a faulted state by these packets. As this was a lab example, we know that the Cybati micrologix_fault Metasploit module was run against the controller. This module sends a command to the PLC to enter it into a faulted state.

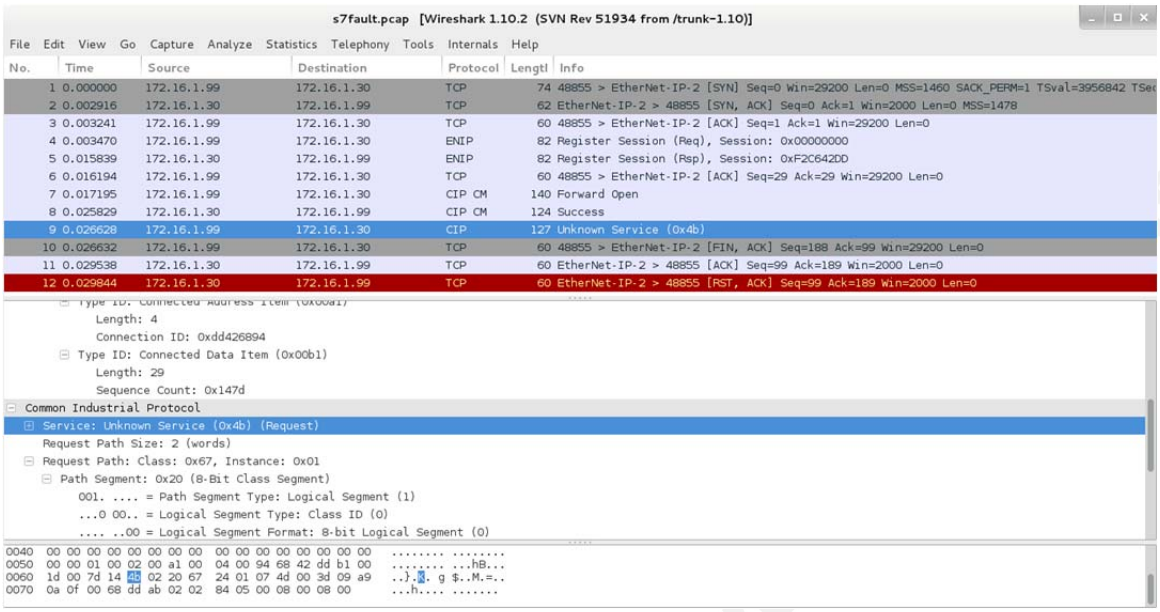


Figure 6: Packet Capture of Exploit Execution

3.2.3. Lessons Learned

An Ethernet monitor located at the PLC in this example enabled significantly more forensic analysis than would have been possible without the monitor. Without the detection of a rogue device on the network, the incident response team would not have been able to identify the malicious behavior. Since the behavior was identified, steps may be taken to eliminate future occurrences of this issue.

3.3. Example 3 – Malicious Code Injection

3.3.1. Scenario

A chemical process malfunctioned, resulting in a hazardous chemical spill. The emergency response team found that a tank had overflowed, and a pump that controlled the tank level by pumping the chemical to another tank was burned out. The emergency response team also noted that the Human-Machine Interface (HMI) for the process had not issued a “High Level” alarm for the tank. Operators stated that the HMI showed the tank level in its middle range and the system operating normally.

Since the “High Level” alarm had not been issued, the emergency response team activated the control system incident response team to ensure the process was safe to restart after the motor had been replaced.

3.3.2. Forensic Analysis

The ladder logic file in the Allen-Bradley MicroLogix 1400 PLC was uploaded to the incident response team's configuration workstation and compared to the baseline program. Note that Allen-Bradley references "upload" and "download" from the PLC's perspective; "upload" is transfer from the PLC, "download" is to the PLC.

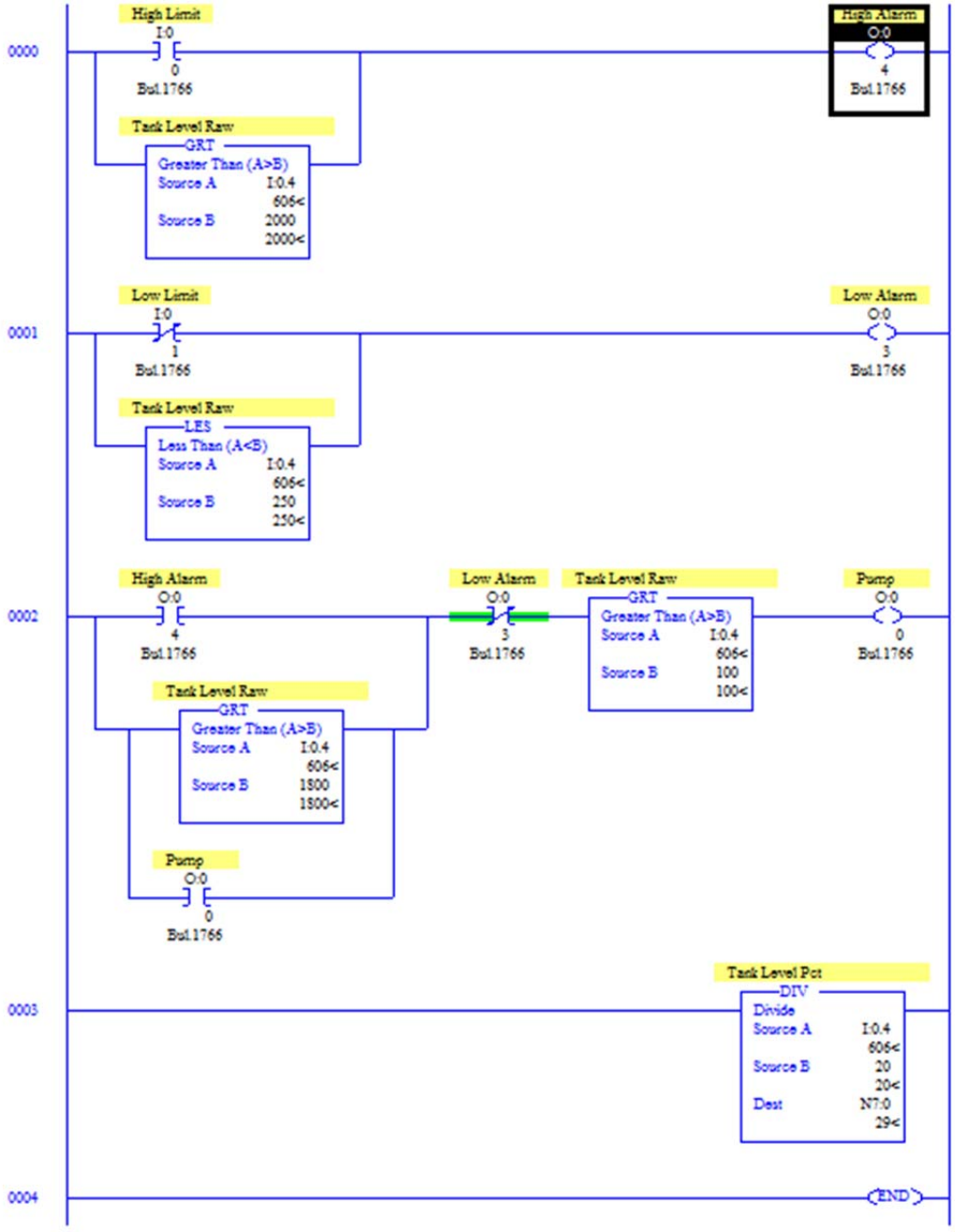


Figure 7: Ladder Logic from Baseline

The process consists of a tank that is filled from a remote source. As the tank level rises, a pump turns on at a pre-set level to move the tank's contents to the next process. When the tank level drops below its lower limit the pump is turned off. High and low level alarms are configured to prevent tank overflow and running the tank empty. If the pump runs dry for too long it will over-speed and burn out.

Figure 7 shows the baseline ladder logic program from the PLC's change management system. Rung 0 turns on the "High Alarm" indicator when the "High Limit" sensor activates or when the raw tank level exceeds 2000 units. Rung 1 turns on the "Low Alarm" sensor when the "Low Limit" sensor deactivates (the tank level drops below the sensor) or the raw tank level falls below 250 units. Rung 2 turns on the pump if a "High Alarm" occurs or the raw tank level exceeds 1800 units. The pump will run until the "Low Alarm" is triggered or the raw tank level drops below 100 units. Rung 3 converts the raw tank level to a percentage by dividing the raw units by 20 and storing the result in an integer for use by the HMI.

Figure 8 shows the ladder logic at the time of the incident. It has been greatly modified from the baseline program. Rung 4 now has a counter that will trigger after the pump runs a set number of times. This will delay execution of the modified logic until the pump runs for the pre-set number of times. Rungs 0 and 1 now are turned off when the counter reaches its pre-set count, disabling the high and low alarms. The counter also turns off the pump until it is turned back on by the timer in Rung 7. This will cause the tank to overflow before the pump turns on. After the timer in Rung 7 turns the pump on, it will run until the timer in Rung 8 expires. This will make the pump run on a dry tank, burning it out. After the Rung 8 timer expires, the counter and both timers are reset by Rung 9, returning the PLC to apparently normal operation. During the operation of the attack, both high and low alarms are disabled and the tank level used by the HMI is set to 37% in an attempt to mask the true activity of the system from the human operator watching the HMI.

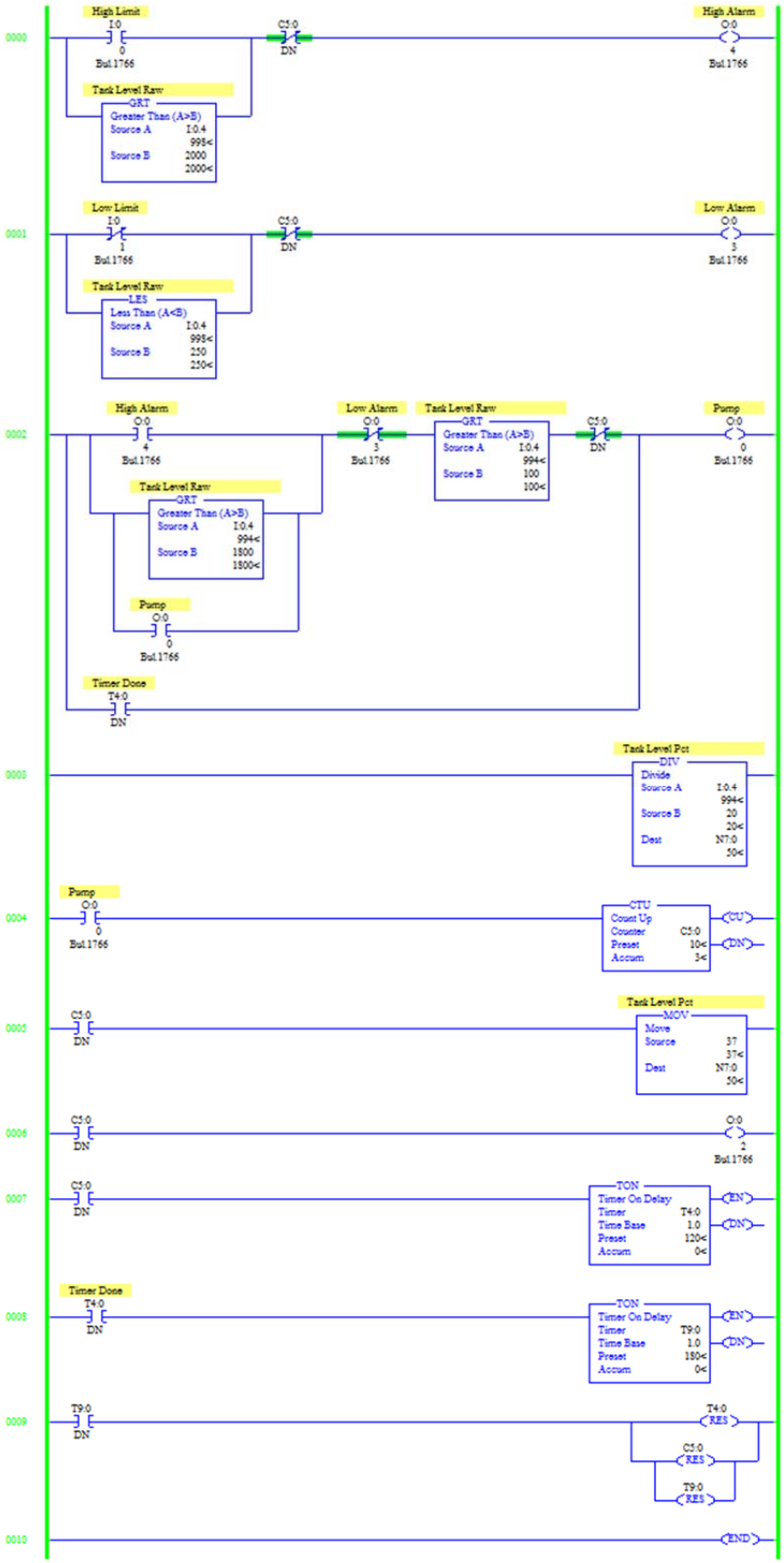


Figure 8: Ladder Logic Uploaded from PLC after Incident

Lew Folkerth, lew.folkerth@rfirst.org

3.3.3. Lessons Learned

Forensic analysis of the incident revealed a deliberate, malicious attack on the process. Had the forensic analysis not been performed, it is likely that the PLC would have resumed control of the process only to cause another incident when the pre-programmed conditions occurred. This is analogous to the performance of Stuxnet (Zetter, 2014).

4. Conclusion

With preparation, it is possible for forensic analysis of control systems to yield usable results. Network monitoring, baseline configurations, and other tools permit the success of the forensic process.

Incident response teams must be familiar with control systems, their protocols, and their programming languages. These can be greatly different from those encountered in an IT environment.

Looking forward, these tools could be supplemented by memory capture capability for PLCs and other field devices. This capability should encompass both running process state and firmware.

5. References

Babcock, N. (2009). *PLC Programming with RSLogix 5000*. Retrieved from http://www.comptechweb.com/images/jr/Challenge/PLCProgrammingwithRSLogix5000.pdf. Last Accessed: September 17, 2015.

Department of Energy. (2012, July). *Electric disturbance events (OE-417)*. Retrieved from Office of Electricity Delivery & Reliability: https://www.oe.netl.doe.gov/oe417.aspx. Last Accessed: September 13, 2015.

Fabro, M., & Cornelius, E. (2008). *Recommended practice: Creating cyber forensics plans for control systems*. DHS Control Systems Security Program. Retrieved from https://ics-cert.us-cert.gov/sites/default/files/recommended_practices/Forensics_RP.pdf. Last Accessed: December 16, 2014.

Fehr, R. (2003, December 1). *The basics of ladder logic*. Retrieved from Electrical Construction & Maintenance Magazine: http://ecmweb.com/archive/basics-ladder-logic. Last Accessed: August 31, 2015.

Fieldbus Foundation. (2014). *Guide to implementing foundation h1 field devices*. Retrieved from Fieldbus Foundation: http://www.fieldbus.org/images/stories/enduserresources/technicalreferences/documents/wp_implement_h1_field_devices_softing.pdf. Last Accessed: August 24, 2015.

Fletcher, D. (2015). *Forensic timeline analysis using Wireshark*. SANS. Retrieved from https://www.sans.org/reading-room/whitepapers/ forensics/forensic-timeline-analysis-wireshark-giac-gcfa-gold-certification-36137. Last Accessed: August 14, 2015.

Henry, P. (2009, September 12). *Best practices in digital evidence collection*. Retrieved from SANS Digital Forensics and Incident Response Blog: https://digital-forensics.sans.org/blog/2009/09/12/best-practices-in-digital-evidence-collection/. Last Accessed: September 17, 2015.

Horkan, M. (2015). *Challenges for IDS/IPS deployment in industrial control systems*. SANS. Retrieved from https://www.sans.org/reading-room/whitepapers/ICS/challenges-ids-ips-deployment-industrial-control-systems-36127. Last Accessed: August 23, 2015.

Kral, P. (2011). *The incident handlers handbook*. SANS. Retrieved from https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901. Last Accessed: August 27, 2015.

Langner, R. (2012). *Robust control system networks: How to achieve reliable control after Stuxnet*. New York, NY: Momentum Press. ISBN: 978-1-60650-300-3.

Lee, R. M. (2013). *SCADA and me*. Birmingham, MI: IT-Harvest Press. ISBN: 978-1491275122.

Macaulay, T., & Singer, B. (2011). *Cybersecurity for industrial control systems: SCADA, DCS, PLC, HMI, and SIS*. Boca Raton, FL: CRC Press. ISBN: 978-1-4665-1610-6.

ODVA. (2008). *EtherNet/CIP - CIP on Ethernet technology*. Retrieved from ODVA: http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00138R3_CIP_Adv_Tech_Series_EtherNetIP.pdf. Last Accessed: August 27, 2015.

OSHA. (2002). *OSHA fact sheet: lockout/tagout*. Retrieved from OSHA: https://www.osha.gov/OshDoc/data_General_Facts/factsheet-lockout-tagout.pdf. Last Accessed: September 6, 2015.

OSHA. (2015, September 17). *OSHA 3151-12R: Personal protective equipment*. Retrieved from OSHA: https://www.osha.gov/Publications/osha3151.html. Last Accessed: September 17, 2015.

Peterson, D. (2013, November 4). *Insecure by design / secure by design*. Retrieved from Digital Bond: http://www.digitalbond.com/blog/2013/11/04/insecure-by-design-secure-by-design/. Last Accessed: September 13, 2015.

Rockwell Automation. (2014). *MicroLogix 1400 programmable controllers: Instruction set reference manual*. Retrieved from Rockwell Automation:

http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1766-rm001_-en-p.pdf. Last Accessed: February 9, 2015.

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., & Hahn, A. (2015). *SP 800-82 Revision 2, Guide to industrial control systems (ICS) security*. Gaithersburg, MD: NIST. Retrieved from NIST: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf. Last Accessed: August 23, 2015.

Wilhoit, K. (2013). *Who's really attacking your ICS equipment?* Trend Micro Incorporated. Retrieved from Trend Micro: http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-whos-really-attacking-your-ics-equipment.pdf. Last Accessed: December 16, 2014.

Witter, F. (2001). *Legal aspects of collecting and preserving computer forensic evidence*. SANS. Retrieved from http://www.giac.org/paper/gsec/636/legal-aspects-collecting-preserving-computer-forensic-evidence/101482. Last Accessed: September 17, 2015.

Wright, C. (2013). *Forensics Management*. In R. Radvanovsky, & J. Brodsky (Eds.), Handbook of SCADA/control system security (pp. 143-176). Boca Raton, FL: CRC Press. ISBN: 978-1-4822-0939-6.

Wu, T., Pagna Disso, J. F., Jones, K., & Campos, A. (2013). *Towards a SCADA forensics architecture*. Proceedings of the 1st International Symposium for ICS & SCADA Cyber Security Research, (pp. 12-21). Retrieved from The Chartered Institute for IT: http://ewic.bcs.org/upload/pdf/ewic_icscsr13_paper2.pdf. Last Accessed: February 14, 2015.

XJTAG. (n.d.). *High-level guide to JTAG*. Retrieved from XJTAG: http://www.xjtag.com/support-jtag/jtag-high-level-guide.php. Last Accessed: September 17, 2015.

Young, S. (2013). *Incident response and SCADA*. In R. Radvanovsky, & J. Brodsky (Eds.), Handbook of SCADA/control system security (pp. 129-142). Boca Raton, FL: CRC Press. ISBN: 978-1-4822-0939-6.

Zetter, K. (2014). *Countdown to zero day: Stuxnet and the launch of the world's first digital weapon.* New York, NY: Crown Publishers. ISBN: 978-0-7704-3617-9.

Acknowledgement

The author wishes to thank Matt Luallen of Cybati, Inc., for the use of an Industrial Cybersecurity Kit as the foundation for the lab work of this paper, and for his advice along the way.