# Mode Matters: Monitoring PLCs for Detecting Potential ICS/OT Incidents

Author: Michael Holcomb, mike@mikeholcomb.com
Advisor: *Tanya Baccam*

Accepted: *January 23rd, 2024*

Abstract

There is a blind spot regarding cyber security in many Industrial Control Systems (ICS) and Operational Technology (OT) networks that support the world around us – in power plants for electricity, water treatment plants for safe drinking water, and railways for safe transportation. Many owners and operators of such environments remain unaware that Programmable Logic Controllers (PLCs) are vulnerable to cyber-attacks, just like their IT counterparts. It is critical that plant operators not only understand how each of their PLC types function but that each is consistently monitored for changes that signal a potential issue is occurring. Such problems could threaten the physical safety of onsite personnel, the surrounding environment, or downtime for the operation. While platforms exist to perform such monitoring, many are considered unaffordable by today's small- to medium-sized environments. As an alternative, some environments might choose to have personnel walk the site to physically examine PLCs, an action that could put those team members in harm's way. This research will help provide a basic framework and sample tool for remotely monitoring PLCs to eliminate such a safety risk.

# 1. Introduction

Most people would likely survive if the power went out for a few hours. However, what if a few hours without power becomes a few days? Or a few months? While cyber-attacks continually threaten traditional IT environments, the world of Industrial Control Systems (ICS), more generally referred to as Operational Technology (OT), is also under attack, as highlighted in Dragos' annual ICS/OT Cybersecurity Year in Review (2022). Examples of such OT environments under attack include but are not limited to, power plants, railways, petrochemical refineries, and water treatment facilities.

These facilities are controlled by Programmable Logic Controllers (PLCs) and other controller types. A PLC is a computer designed to control processes in the real world. Brooks and Craig (2022) state that PLCs are "the preferred local control device in modern industrial processing and utility environments." A typical example of a PLC is a modern thermostat in homes and offices. The thermostat can measure the temperature at a location using a sensor. If the temperature rises above the selected setpoint, the thermostat sends an electrical signal to the air conditioning unit to turn on.

The PLC has a CPU, memory, and limited storage capacity. It has its operating system and runs code. Even though the PLC might look physically different from a standard workstation or laptop, it shares most of the same components – and many of the same vulnerabilities as well as some of its own. Additionally, as Ackerman (2021) highlighted, such controls are now networked over TCP/IP, making them remotely accessible. Also, the data transmitted by such controls "can be easily interpreted because the controls and automation protocols that ride on top of IP and TCP are just about all cleartext protocols (Ackerman, 2021).”

Unfortunately, just like other computers in the IT world, these computers are vulnerable to attack. Attackers can target these systems as well as the systems and processes connected to the controllers. If a power plant that provides electricity to 250,000 homes in the middle of winter is taken offline by a cyber-attack similar to an event that took place in the Ukraine in 2015 (Greenberg, 2020), there could be severe ramifications.

As highlighted in the MITRE ATT&CK Framework for ICS (The MITRE Corporation, 2023), several attacks could be leveraged by attackers who take advantage of a PLC's Operational Mode. These include performing reconnaissance against a PLC to remotely determine its functional status (T0868), uploading malicious PLC code or firmware (T0858) and conducting successful Denial of Service attacks exploiting a PLC's firmware update mode (T0800). Additional techniques highlighted include the ability for attackers to remotely enumerate a PLC for additional information about itself (T0888) to help further the attacker's objectives.

In incidents like the 2017 Trisis event, attackers compromised controllers over the network in control system environments. As pointed out by Steve Mustard, "Attackers would not have been able to modify the code on the safety controller in a Middle East petrochemical facility had the facility personnel kept the physical key switch on that device in “Run” mode (Mustard, 2022).” A simple solution could have prevented this incident by setting the controller's key switch to “Run” Mode.

Most PLCs have different “Operational” modes, such as "Run" and "Stop," which can help limit cyber-attack risks. The representation of a key switch typically controls these operational modes – physically or via software. A PLC might include a mode for programming itself or allowing the uploading of firmware. A second mode places the PLC into "Read-Only" Mode, also referred to as “Run” mode, which prevents programming and firmware changes from being made on some PLCs. If programming changes and firmware updates cannot be made, the PLC is protected against remote attacks. (Stauffer, 2023)

Vendors implement operational modes differently, so cyber security for PLCs is more complex than ensuring the PLC is always in “Run” mode. Every PLC brand is unique, with different operational modes that can function very differently from their counterparts from other providers. Operators must understand these differences. Moreover, at the same time, they must also monitor when PLCs are taken out of “Run” Mode to help identify potential operational and security issues in the environment.

# 2. Research Method

Several common PLCs were purchased and deployed in a lab environment to help determine if different operational modes can prevent unauthorized changes from being made to a PLC and how best to monitor for potential changes that could leave the asset in a vulnerable state. Particular attention was paid to determining the operational modes for each PLC and how the asset operates in each mode. Each PLC had its different operational modes listed, reviewed, and tested in the lab to determine if changes could be made to the PLC while in “Run” or "Read-Only" Mode.

Following the operational mode testing, the PLCs were examined to determine how each might remotely advertise the current operational mode. If its operational mode was remotely advertised, an application could be developed to monitor when the PLC is taken out of “Run” Mode, alerting operators of a potential security or operational issue. ChatGPT was used to create scripts and applications related to this monitoring functionality where possible.

Sample code generated by ChatGPT in support of this project can be found at github.com/utilsec.

## 2.1. Research Lab Environment

The lab environment was established with PLCs from several common industry vendors found in ICS/OT environments in North America (Figure 1). An Engineering Workstation (EWS) running Windows 11 was deployed to configure, program, and test each PLC. The workstation was also used to analyze network traffic between network hosts. All hosts were interconnected with an unmanaged network switch. Network traffic was continually captured with Wireshark during testing. For DHCP services, the Tftpd64 application was installed on the EWS to provide IP addresses to PLCs, which required a dynamic IP address assignment during initial setup.

The lab environment was established with an engineering workstation (192.168.100.100) and PLCs from two different vendors commonly found deployed in North American OT environments:

- ClickPLUS PLC (192.168.100.200) from Automation Direct

- Micro820 PLC (192.168.100.210) from Allen-Bradley / Rockwell Automation

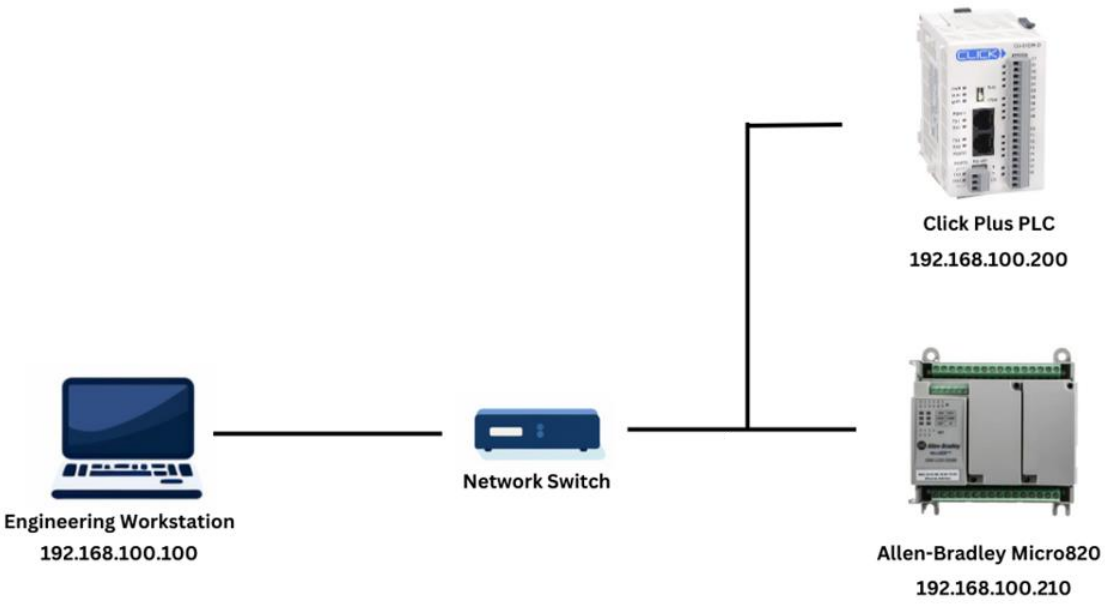


Figure 1. Research Lab Environment

## 3. Findings and Discussion

Once the lab environment was established with all three PLCs initially programmed and running, the next step in the research was to review the operational mode of the PLCs, starting with the CLICK PLUS PLC. As highlighted on the ‘Secure PLC Coding Practices: Top 20 List’, when it comes to securing PLCs, it is essential to "track operating modes" and to "keep the PLC in “Run” mode.” There is also a focus on raising an alert for when a PLC is switched out of “Run” mode to alert operators. (admeritia GmbH, 2022).

### 3.1. Review of the CLICK PLUS PLC’s Operational Modes

Due to its popularity as a fully functional PLC at a relatively affordable price, the CLICK PLUS PLC from Automation Direct was examined first. The CLICK PLUS PLC, previously configured with an IP address of 192.168.100.200, was connected to the research lab network.

A physical review of the CLICK PLUS PLC, along with a review of the PLC's vendor documentation (Automationdirect.com, 2023), determined that only two operational modes exist for the CLICK PLUS PLC:

- "Run" mode
- Stop Mode

A physical DIP switch and the CLICK Programming Software client installed on an engineering workstation (Figure 2) control the two operational modes on the PLC. If there is a conflict between the operating mode settings between the physical DIP switch and the software selection, the hardware switch selection takes precedence.

As highlighted by Hoffman and Cedillo (2023), 'aside from control system and automation support roles, the position of these switches was relatively unknown or largely unconsidered to operational personnel.' There needs to be more awareness of the importance of key switches and the operational modes of PLCs. While Hoffman and Cedillo's (2023) research previously focused on monitoring the state of a key switch, the focus here is monitoring the operational mode itself.

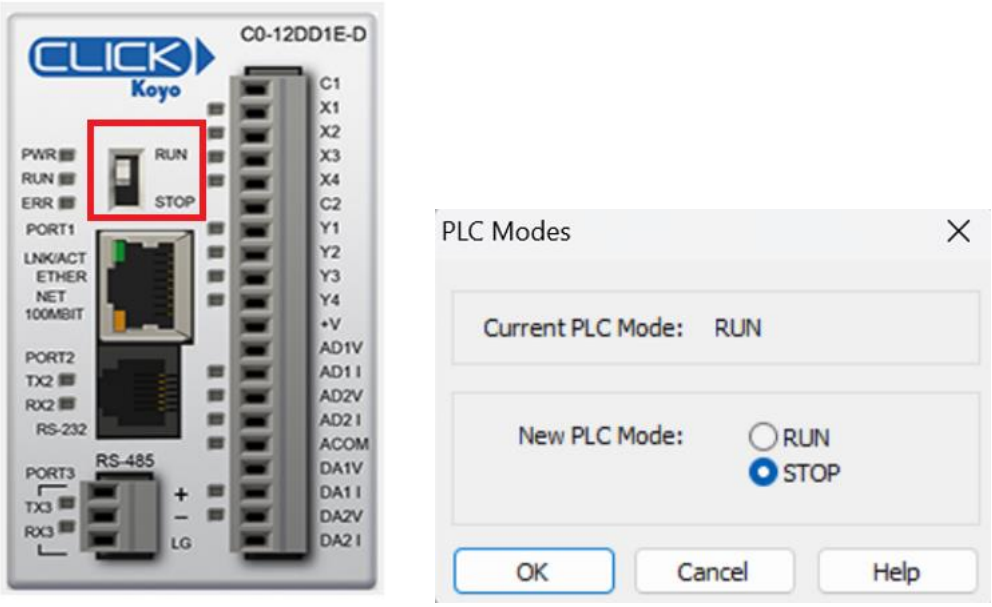


Figure 2. Physical Switch and Software Options for Changing Operational Modes

"Run" Mode for a CLICK PLUS PLC allows the PLC to operate and execute the ladder logic programming code in memory. In addition, updates can be made to the PLC's inputs and outputs, allowing for the collection of sensor data and the PLC to send signals to the systems it controls. Though not immediately apparent from physically examining the PLC or reading the user manual, there is a third option, one that allows PLC programming changes to be made even while the PLC is in "Run" mode.

## 3.2. Can CLICK PLC Changes be Made in "Run" Mode?

"Stop" Mode is the opposite of "Run" Mode, in which the CLICK PLUS PLC does not operate, execute any ladder logic programming, or process any input or output activity. One everyday use of "Stop" Mode on the CLICK PLUS PLC would be upgrading the system's firmware. A firmware upgrade was attempted while the CLICK PLUS PLC was in "Run" mode, and the following message was displayed (Figure 3).

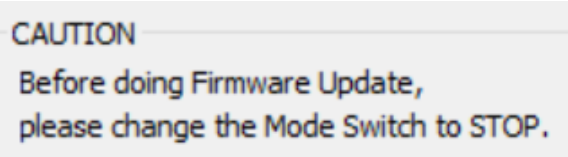


Figure 3. PLC Warning Requiring STOP Mode for Firmware Update

That said, it was determined that the PLC does not need to be placed in "STOP" mode to make programming changes, only to upgrade firmware. Once the system receives the payload, it can examine the flag to determine whether the CLICK PLUS PLC is in "Run" mode as seen in Figure 4 below.

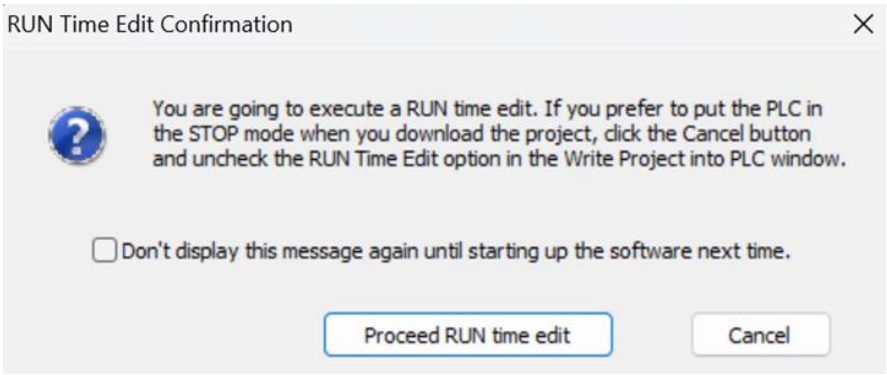


Figure 4. RUN Time Edit Confirmation Screen

When changing the ladder logic programming running on the PLC while in "RUN" mode, the option was presented to place the PLC into 'RUN Time Edit' (Figure 5). This mode allows the PLC programming to be updated while the asset is in "Run" mode if the user knows the administrator password. In the case of the CLICK PLUS PLC, keeping the operational mode in "Run" mode does not ensure that PLC programming cannot be altered by updating its ladder logic code, though it does enforce that the firmware of the asset cannot be changed.


Figure 5. RUN Time Edit Selection Option to Bypass "Run" Mode

While the "Run" mode of a CLICK PLUS PLC was determined not to prevent PLC programming changes, it would prevent unauthorized firmware uploads. Therefore, it was determined that it would be worth monitoring the operational mode of the CLICK PLUS PLC for when the PLC might be taken out of "Run" mode. If the CLICK PLUS PLC is taken out of "Run" mode, it could indicate that an attacker is attempting to upload malicious firmware or make another unauthorized change. Any unexpected change in operational status, such as a PLC being taken out of "Run" mode outside of an authorized change window, should be investigated to rule out any associated operational and security issues. Because monitoring the operational status of the PLC would be valuable from a security monitoring standpoint, the next step in the research was to determine if the current operational status of the CLICK PLUS PLC could be remotely enumerated continually.

## 3.3. Determining if a CLICK PLC Mode Can Be Enumerated

Once a PLC is on the network, it must be connected to the appropriate management software client running on an engineering workstation. In the case of the CLICK PLUS PLC line, the CLICK Programming Software client (v3.41) was run on the EWS. Once launched, the application allows the user to "Connect to PLC." Selecting this option results in the "Connect to CLICK PLUS PLC" window appearing (see Figure 6

below). As seen in Figure 6, the “Run” mode of the PLC is automatically displayed along with other properties of the PLC (e.g., IP address, subnet mask, MAC address, firmware version). The software client requires no authentication to receive or access this information.

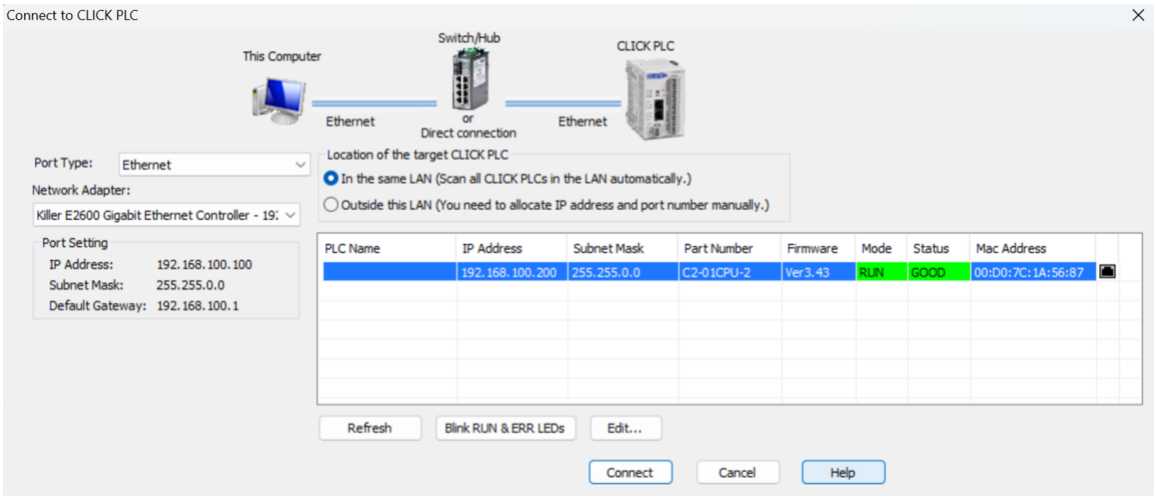


Figure 6. Click Programming Software PLC Query Screen

With the current “Run” mode of the PLC automatically being advertised, the Wireshark packet capture was stopped. The communication between the PLC the EWS was reviewed to determine how the current operational mode was advertised by the PLC. Upon review, it was discovered that the PLC status data, as seen in the Figure above, is the result of a process initiated by the EWS. When initiated, the EWS sends out three identical broadcast packets to UDP 25425 from a source port of UDP 2770 (Figure 7).

| Source | Destination | Protocol | Leng | Info |
|---|---|---|---|---|
| 192.168.100.100 | 255.255.255.255 | UDP | 56 | 2770 → 25425 Len=14 |
| 192.168.100.200 | 255.255.255.255 | UDP | 115 | 25425 → 2770 Len=73 |
| 192.168.100.100 | 255.255.255.255 | UDP | 56 | 2770 → 25425 Len=14 |
| 192.168.100.200 | 255.255.255.255 | UDP | 115 | 25425 → 2770 Len=73 |
| 192.168.100.100 | 255.255.255.255 | UDP | 56 | 2770 → 25425 Len=14 |
| 192.168.100.200 | 255.255.255.255 | UDP | 115 | 25425 → 2770 Len=73 |

Figure 7. Packet Capture of the CLICK PLUS PLC Broadcast Discovery Process

This six-step process is called the "CLICK PLUS PLC Discovery Process." The process is comprised of two distinct packets sent three times each. The first broadcast

packet from the engineering workstation is called the "CLICK PLUS PLC Discovery Request." The corresponding response from the CLICK PLUS PLC is called the "CLICK PLUS PLC Discovery Response."

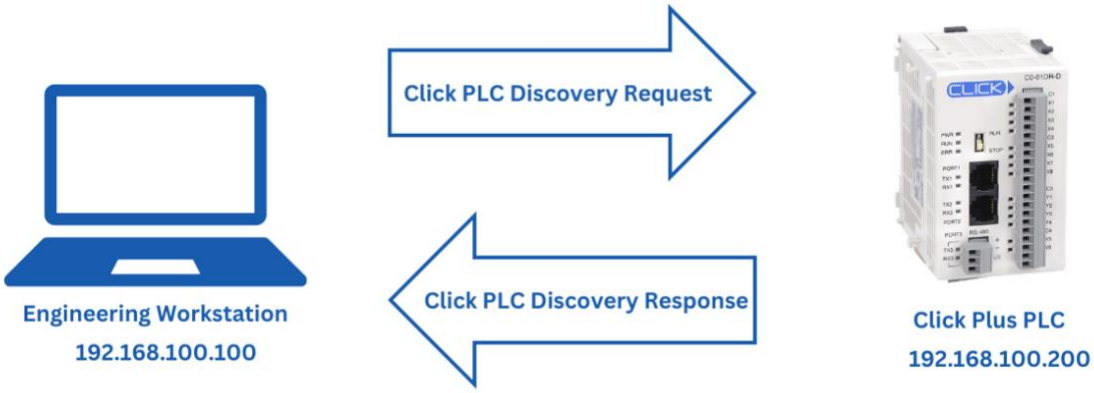


Figure 8. The CLICK PLUS PLC Discovery Process

The Engineering Workstation sends the initial data for the three CLICK PLUS PLC Discovery Request packets, which include a specific 120-bit hex string (‘484b4f50000100edd5040045016680’) in the payload seen in Figure 9 below. This initial inquiry string never changed during testing and appears constant. Another constant string that the EWS software can send sends the command to the PLC to blink its RUN and ERR lights, which would help a physically present technician readily identify the PLC from others. The 192-bit specific hex string sent for this command is 4b4f500001006a570e0045016643c0a864c800d07c1a5687.

```
> Ethernet II, Src: Dell_1b:01:17 (04:bf:1b:1b:01:17), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 192.168.100.100, Dst: 255.255.255.255
v User Datagram Protocol, Src Port: 2770, Dst Port: 25425
    Source Port: 2770
    Destination Port: 25425
    Length: 22
    Checksum: 0x32eb [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  > [Timestamps]
    UDP payload (14 bytes)
v Data (14 bytes)
    Data: 4b4f50000100edd5040045016680
    [Length: 14]
```

Figure 9. Payload Data of the CLICK PLUS PLC Discovery Request Packet

The PLC, connected to the same local network as the EWS, receives the broadcast requests. In the case of the initial 120-bit discovery request, the PLC will respond with its own specially constructed 584-bit hex string. This response contains the data used to provide the information displayed in the earlier discovery screen. In this example, the whole hex string transmitted by the PLC was 4b4f50eb01004d093f00450166c110000100c0a864c8ffff00000000000000d07c1a568700000000000000000000000000000000000000000000000000000000000000c7032b00800001.

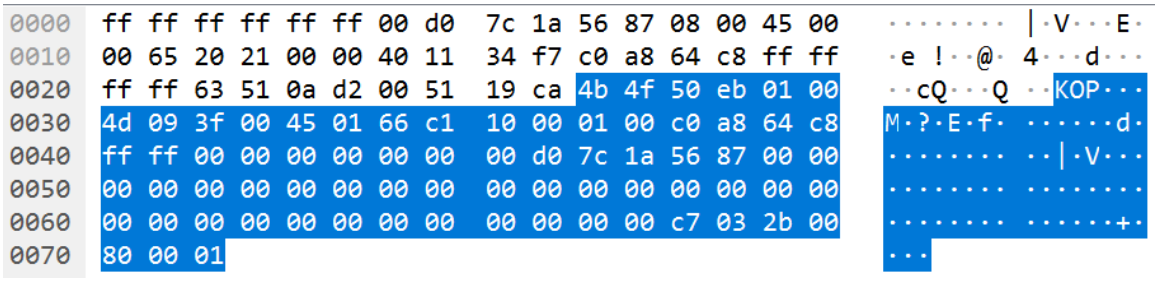


Figure 10. Payload Data of the CLICK PLUS PLC Discovery Response String

Reviewing the 584-bit response string and comparing it with the PLC information seen on the first screen, the response string needed to be deconstructed to determine how the current “Run” mode status of the PLC is being communicated.

The response string was comprised of many fields. For example, the first field, always constant, comprises the first three hex values ('4b4f50'), which translates to KOP in ASCII text. KOP stands for 'Kontaktplan,' the German word for ladder logic, the programming language used on the CLICK PLUS PLC. These values appear as the initial values in the Discovery Request and the Discovery Response payload. Additional fields are distinguishable based on the information displayed in the CLICK programming software (e.g., IP address, subnet mask, MAC address).

To help further narrow down which portion of the response relays the “Run” mode status, the key switch on the CLICK PLUS PLC was moved to the STOP position. The PLC was again remotely queried for its status (Figure 11). While all other fields in the response are the same, the “Run” mode field is now responding as 'STOP' in the GUI.

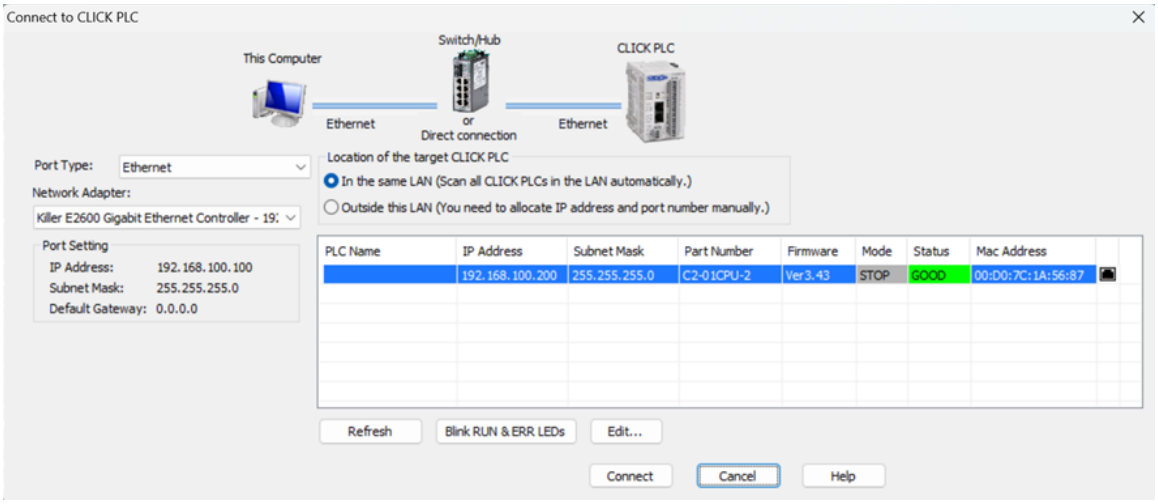


Figure 11. CLICK PLUS PLC Broadcast Discovery Results Displayed

The data returned in the packet captures were reviewed to determine which value would represent the “Run” mode. Additional captures were taken over several days, helping to determine fields that might contain other valuable data. For example, a name (ABCDEFGHIJKLMNOPQRSTUVWX) was assigned to the PLC to determine which part of the response payload stored the provided PLC name.

Overall, 16 unique fields were discovered throughout the research process (Figure 12). The following fifteen unique fields were identified:

| Field # | Constant? | Length | Determined Purpose |
|---|---|---|---|
| 1 | Constant | 24-bit | Identifies itself as a Click PLC packet |
| 2 | Variable | 8-bit | Unknown |
| 3 | Constant | 16-bit | Unknown |

| | | | |
|---|---|---|---|
| 4 | Constant | 16-bit | Unknown |
| 5 | Constant | 80-bit | Unknown |
| 6 | Variable | 32-bit | PLC IP address |
| 7 | Constant | 16-bit | Broadcast destination (255.255.255.255) |
| 8 | Variable | 16-bit | PLC Subnet Mask |
| 9 | Constant | 32-bit | Unknown |
| 10 | Constant | 48-bit | PLC MAC address |
| 11 | Variable | 192-bit | PLC name |
| 12 | Constant | 48-bit | Unknown |
| 13 | Constant | 8-bit | Unknown |
| 14 | Variable | 16-bit | Indicates firmware version of the PLC |
| 15 | Variable | 16-bit | Indicates the operational status of the PLC |
| 16 | Constant | 16-bit | Unknown |

Figure 12. CLICK PLUS PLC Discover Response Request Fields

Running through different variations of the two principal operational modes for the PLC and comparing the packet capture results, Field #15, as indicated above in Figure 12, was discovered to indicate whether or not the PLC was in “Run” mode. Not all fields had an associated purpose or meaning discovered during this research though further testing and research could reveal additional purposes.

Only one value would change through all the captures and combinations of “Run” mode settings, along with different dates and settings, indicating whether the essential switch mode for the CLICK PLUS PLC is in “Run” or “Stop” mode. This value was in the second to last field near the end of the returned hex string. Additional testing was done using the physical switch and the software client to switch the PLC between RUN and STOP modes.

Field #15 was determined to have the three following values during testing:

- 03c0 = The PLC is in “Run” mode
- 0080 = The PLC was placed into STOP mode by the hardware switch
- 00c0 = The PLC was placed into STOP mode by the EWS software

If both the hardware and software are configured for STOP mode, Field #15 indicates '0080', which aligns with the hardware setting overriding the software setting. Other modes and values not discovered during the research testing could exist.

## 3.4. Remotely Monitoring for “Run” mode Change on the CLICK PLUS PLC

Once a field in the Discovery Response Request packet had been determined to indicate the PLC's current operational status, a method to remotely monitor this value needed to be developed. This method would elicit a response from the CLICK PLUS PLC by using Scapy to duplicate the CLICK Programming Software broadcast request initially sent to gather this information. Scapy is a software utility capable of crafting unique network packets to be transmitted on the network. It can also be used to capture network traffic.

While the CLICK PLUS PLC and associated software clients use broadcast traffic to communicate, further research testing will focus on limiting the amount of broadcast traffic on the network. Broadcast traffic can be seen as detrimental to an OT network and have potentially unforeseen consequences.

In this case, Scapy generated a network packet directly to the CLICK PLUS PLC address (192.168.100.200) rather than the broadcast address (255.255.255.255).

```
>>> ip_packet = IP(src="192.168.100.100", dst="192.168.100.200")
>>> udp_packet = UDP(sport=2770, dport=25425)
>>> hex_data = "4b4f50000100edd5040045016680"
>>> data = bytes.fromhex(hex_data)
>>> packet = ip_packet / udp_packet / Raw(load=data)
>>> send(packet)
.
Sent 1 packets.
```

Figure 13. Scapy Commands to Emulate CLICK PLUS PLC Discovery Request Packet

As can be seen in the Wireshark packet capture below, the initial discovery request packet sent directly to the CLICK PLUS PLC elicits the same response from the PLC as earlier. In this instance, the response is still sent as a broadcast. Unlike the default program, which sends three initial discovery packets as broadcasts, only a single unicast packet was sent directly to the PLC, which in return elicited a single broadcast packet. The process reduced broadcast packets from six to one, an 84% reduction.

| Source | Destination | Protocol | Leng | Info |
|---|---|---|---|---|
| 192.168.100.100 | 192.168.100.200 | UDP | 56 | 2770 → 25425 Len=14 |
| 192.168.100.200 | 255.255.255.255 | UDP | 115 | 25425 → 2770 Len=73 |

Figure 14. CLICK PLUS PLC Discovery Process Initiated by Scapy Crafted Packet

Upon checking the returned broadcast announcement, the packet data still contains the expected payload with unencrypted details about the CLICK PLUS PLC.

```
0000  ff ff ff ff ff ff 00 d0  7c 1a 56 87 08 00 45 00   ........ |·V···E·
0010  00 65 20 28 00 00 40 11  34 f0 c0 a8 64 c8 ff ff   ·e (··@· 4···d···
0020  ff ff 63 51 0a d2 00 51  19 ca 4b 4f 50 eb 01 00   ··cQ···Q ··KOP···
0030  4d 09 3f 00 45 01 66 c1  10 00 01 00 c0 a8 64 c8   M·?·E·f· ......d·
0040  ff ff 00 00 00 00 00 00  00 d0 7c 1a 56 87 00 00   ........ ··|·V···
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 c7 03 2b 00   ........ ......+·
0070  80 00 01                                          ···
```

Figure 15. CLICK PLUS PLC Discovery Response to Scapy Crafted Request

Realizing that UDP port 2770 would be bound to the CLICK Programming Software client and not a viable source port, the source port for the initial broadcast

request was changed to UDP 22770. A high-ordered port would prevent conflicts with other software applications on the Engineering Workstation. New packets were crafted with Scapy with the UDP 22770 source port, and it was discovered that the PLC responds to the originating source port (over a broadcast).

| Source | Destination | Protocol | Leng | Info |
|---|---|---|---|---|
| 192.168.100.100 | 192.168.100.255 | UDP | 56 | 22770 → 25425 Len=14 |
| 192.168.100.200 | 255.255.255.255 | UDP | 115 | 25425 → 22770 Len=73 |

Figure 16. PLC Responding to Discovery Request on Non-Standard Source Port

Now that a response has been successfully elicited without using the CLICK Programming client and from a port that will not conflict with other applications, the next step is to send the packet regularly. For research purposes, the discovery request will be sent every 15 seconds. A Python script was written using ChatGPT (Figure 17).

```
From scapy.all import *
import time

# Define packet parameters
ip_packet = IP(src="192.168.100.100", dst="192.168.100.255")
udp_packet = UDP(sport=2770, dport=25425)
hex_data = "4b4f50000100edd5040045016680"
data = bytes.from hex(hex_data)
packet = ip_packet / udp_packet / Raw(load=data)

# Send the packet every 15 seconds indefinitely
while True:
  send(packet)
  print("Packet sent. Waiting 15 seconds before sending the next one.")
  time.sleep(15)
```

Figure 17. Script for Sending CLICK PLUS PLC Discovery Request Every 15 Seconds

At the same time, a Scapy listener on UDP 22770 will be setup to receive the broadcast response from CLICK PLUS PLC. Once received, the payload can be examined for the flag, determining whether the CLICK PLUS PLC is in “Run” mode.

```
def packet_callback(packet):
   # Check if the packet is the one we are interested in
   if packet. haslayer(IP) and packet[IP].src == '192.168.100.200' and \
     packet.haslayer(UDP) and packet[UDP].dport == 22770:
      # Check if there is a payload
      If packet.haslayer(Raw):
         # Extract the payload
         payload = packet[Raw].load
         # Convert the payload to a hex string
         hex_payload = payload.hex()

         # Get the last eight characters of the hex string
         last_eight_chars = hex_payload[-8:]
         print(f'Last 8 Hex Characters from {packet[IP].src}: {last_eight_chars}")

         # Check the last eight characters of the hex string
         if last_eight_chars == '03c00001':
            print(""“RUN” MODE")
         Else:
            print("INVESTIGATE")

# Start sniffing packets
sniff(iface='Killer E2600 Gigabit Ethernet Controller', prn=packet_callback,
store=0, filter="udp port 22770")
```

Figure 18. Script for Capturing and Parsing Responses

To reduce the amount of screen real estate used, only the last 32 bits of the payload will be displayed rather than the entire 584-bit payload string (Figure 19).

```
Last 8 Hex Characters from 192.168.100.200: 03c00001
RUN MODE
Last 8 Hex Characters from 192.168.100.200: 03c00001
RUN MODE
Last 8 Hex Characters from 192.168.100.200: 03c00001
RUN MODE
Last 8 Hex Characters from 192.168.100.200: 03c00001
RUN MODE
Last 8 Hex Characters from 192.168.100.200: 00800001
INVESTIGATE
```

Figure 19. Screen Output from Capturing and Parsing PLC Responses

Now that a listening service has been configured to receive and parse broadcast advertisements from the CLICK PLUS PLC, we want to be able to display this information for review. The previous script was updated to parse the broadcast from the CLICK PLUS PLC and write the status ("“RUN” MODE' or 'INVESTIGATE') to a local file (status.txt).

A Python Flask web application was then created to display the status for the CLICK PLUS PLC. The operational Mode is changed using both the hardware switch and software option to verify that the webpage accurately reflects the operational status of the PLC. In “Run” mode, the status page displays the words' “RUN” MODE' and a green checkmark.


Figure 20. PLC Status Monitoring Page Showing PLC in “Run” mode

When the CLICK PLUS PLC is taken out of “Run” mode by any method, the PLC monitoring status page shows the word 'INVESTIGATE' in red font. Three red exclamation points are also shown on the image representing the CLICK PLUS PLC.


Figure 21. PLC Status Monitoring Page Showing PLC Not in “Run” mode

When an operator notices that a PLC has been placed in ‘INVESTIGATE’ status, the operator needs to immediately determine the cause of the PLC no longer being in “Run” mode. Timely identification and remediation of any operational or security issue is essential to preventing incidents that threaten physical or environmental safety, as well as the operations of the site.

## 3.5. Review of the Allen-Bradley/Rockwell Automation Micro820 PLC

In addition to the CLICK PLUS PLC, the Micro820 PLC from Allen-Bradley (now Rockwell Automation) was reviewed. As described in the Micro820 Programmable Controllers User Manual (Rockwell Automation, 2023), the PLC has the following operational modes:

- Program Mode
- “Run” mode
- REM “Run” mode (Remote)
- “Run” mode Change (RMC)
- “Run” mode Configuration Change (RMCC)

Each of these last two operational modes allows changes to be made to the PLC while it is in “Run” mode. Like the CLICK PLUS PLC, ensuring that the Micro820 is

always in “Run” mode does not mean its running code cannot be updated. Even in “Run” mode, it could be successfully altered or compromised.

Like the CLICK PLUS PLC, the Micro820 operational mode can be controlled by a hardware component or through software. The hardware component is not a physical switch on the PLC itself but comes as an additional module (2080-REMLCD) that can be connected to the PLC. This additional model was not obtained for the research project.

The software client used to manage the Micro820 is Connected Components Workbench. CCW can discover Micro820 PLCs on the local network via a broadcast request like the CLICK PLUS PLC and its client software. Unlike the CLICK PLUS PLC, the Micro820 does not automatically advertise its current operational mode in the initial discovery communication between the client and PLC.

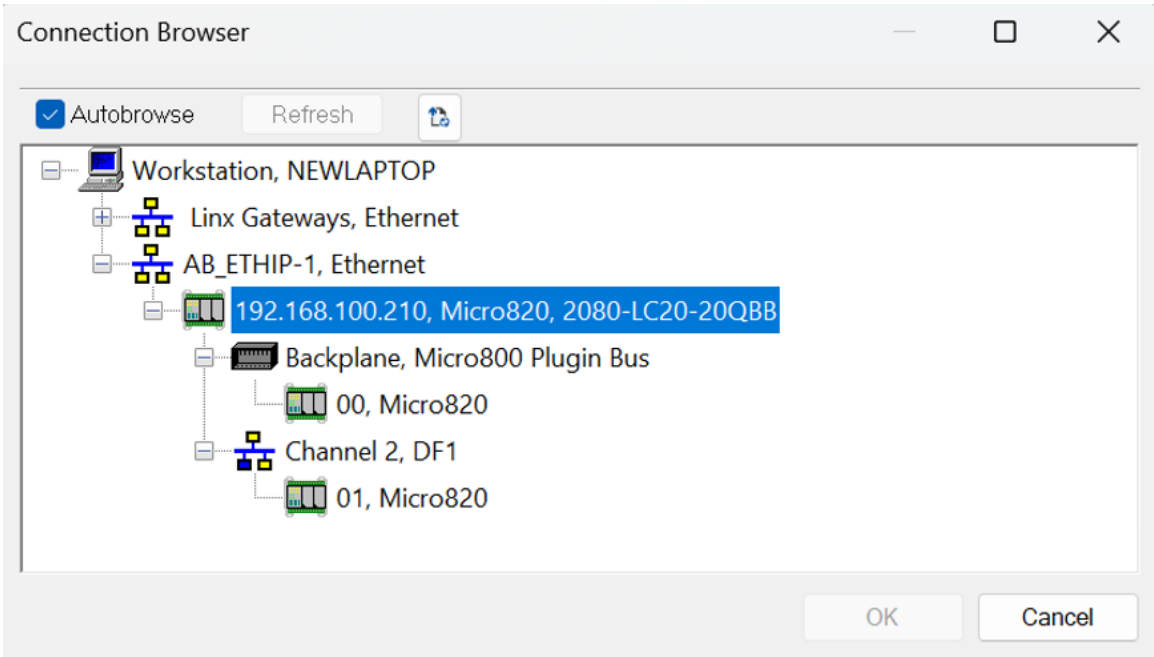


Figure 22. Allen-Bradley Micro820 PLC Discovery in CCW

Once the CCW software is used to connect to the Micro820 PLC, the current operational Mode for the PLC is displayed along with a toggle switch to change between Run and Program modes.

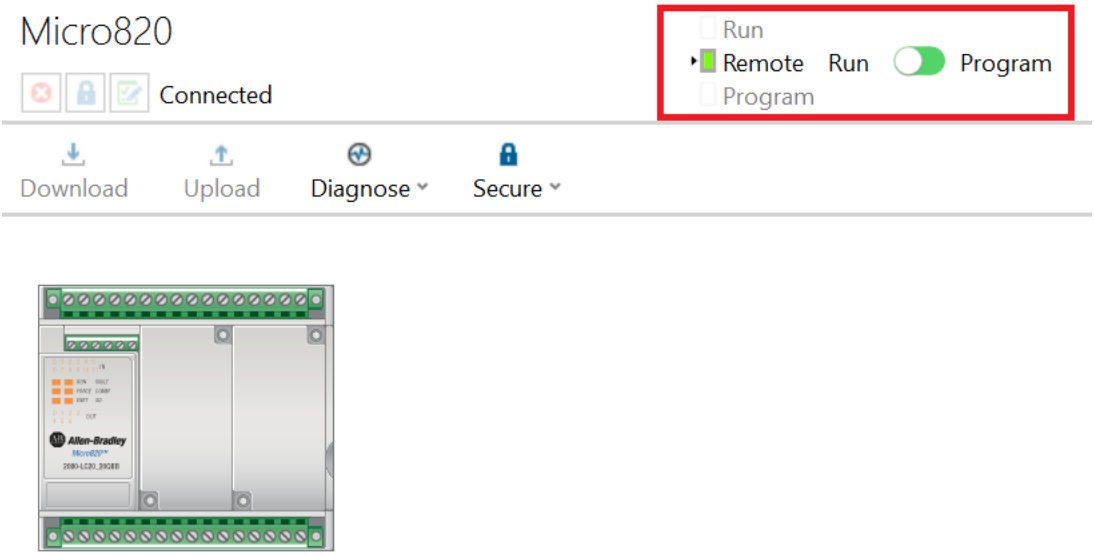


Figure 23. CCW Software Switch for Changing Operational Mode

As seen in the associated packet capture, the information displayed in the CCW Connection Browser is provided by the Micro820 PLC over EtherNet/IP on TCP 44818.

| Source | Destination | Protocol | Leng | Info |
|---|---|---|---|---|
| 192.168.100.100 | 192.168.100.210 | CIP | 106 | Identity - Get Attributes All |
| 192.168.100.210 | 192.168.100.100 | CIP | 145 | Success: Identity - Get Attributes All |
| 192.168.100.100 | 192.168.100.210 | CIP | 108 | Identity - Get Attribute Single |
| 192.168.100.210 | 192.168.100.100 | CIP | 129 | Success: Identity - Get Attribute Single |
| 192.168.100.100 | 192.168.100.210 | CIP | 108 | Identity - Get Attribute Single |
| 192.168.100.210 | 192.168.100.100 | CIP | 113 | Success: Identity - Get Attribute Single |

Figure 24. Sample of EtherNet/IP Communication Between CCW and PLC

Bodungen et al. (2017, pp. 149–153) highlight that a Nmap script (snip-info.nse) exists to query ICS/OT hosts over EtherNet/IP, which runs on TCP 44818.

```
C:\Users\micha>nmap 192.168.100.210 -p 44818 --script enip-info.nse
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-07 09:09 Eastern Standard Time
Nmap scan report for 192.168.100.210
Host is up (0.0022s latency).

PORT      STATE SERVICE
44818/tcp open  EtherNet-IP-2
| enip-info:
|   type: Programmable Logic Controller (14)
|   vendor: Rockwell Automation/Allen-Bradley (1)
|   productName: 2080-LC20-20QBB
|   serialNumber: 0x7073d9f6
|   productCode: 181
|   revision: 12.11
|   status: 0x0034
|   state: 0x03
|_  deviceIp: 192.168.100.210
MAC Address: BC:F4:99:00:13:92 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
```

Figure 25. Nmap Script Enumerating Micro820 PLC in “Run” mode

Once the initial enumeration was complete, the Micro820 PLC was placed into Program mode. The Nmap enumeration script was rerun against the PLC. When enumerated in Program mode, the Nmap script shows that the 'state' value has changed from 0x03 to 0x02.

```
C:\Users\micha>nmap 192.168.100.210 -p 44818 -sV -sC
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-06 16:34 Eastern Standard Time
Nmap scan report for 192.168.100.210
Host is up (0.0021s latency).

PORT      STATE SERVICE       VERSION
44818/tcp open  EtherNet-IP-2
| enip-info:
|   type: Programmable Logic Controller (14)
|   vendor: Rockwell Automation/Allen-Bradley (1)
|   productName: 2080-LC20-20QBB
|   serialNumber: 0x7073d9f6
|   productCode: 181
|   revision: 12.11
|   status: 0x0034
|   state: 0x02
|_  deviceIp: 192.168.100.210
MAC Address: BC:F4:99:00:13:92 (Unknown)
```

Figure 26. Nmap Script Enumerating Micro820 PLC in Program Mode

Once the State was determined to reflect the operational Mode of the PLC and the return data could be identified in the response, ChatGPT was used to generate a Python script to enumerate the state value remotely. While the original script it created did not

determine the appropriate location for this data, the script was manually updated to point to the appropriate location.

```
192.168.100.100    192.168.100.210    ENIP     78 List Identity (Req)
192.168.100.210    192.168.100.100    ENIP    133 List Identity (Rsp), 2080-LC20-20QBB
```

Figure 27. ChatGPT Generated EtherNet/IP 'List Identity' Request

```
Item Count: 1
  Type ID: CIP Identity (0x000c)
      Length: 49
      Encapsulation Protocol Version: 1
    > Socket Address
      Vendor ID: Rockwell Automation/Allen-Bradley (0x0001)
      Device Type: Programmable Logic Controller (14)
      Product Code: 181
      Revision: 12.11
      Status: 0x0034
      Serial Number: 0x7073d9f6
      Product Name Length: 15
      Product Name: 2080-LC20-20QBB
      State: 0x03
```

Figure 28. PLC Response to EtherNet/IP ‘List Identity’ Request

Once the state value can be retrieved over the network, the State is written to a local text file, which can be used to determine whether the PLC is in “Run” mode. An Nmap script would suffice, though it requires Nmap to be installed on the engineering workstation. To eliminate the risk of having an authorized Nmap installation in the OT network be used unauthorizedly, a similar method from the CLICK PLUS PLC was used to enumerate the Micro820 PLC operational status.

## 3.6. Bringing It All Together in a Singe Monitoring App

A new web application interface was created to display PLCs' monitoring status and operational modes, especially for large environments that could potentially have dozens, if not hundreds, of PLCs. A simple interface was created with basic asset

information. If a listed PLC came out of "Run" mode, the status would change and raise and alert for an analyst to investigate.

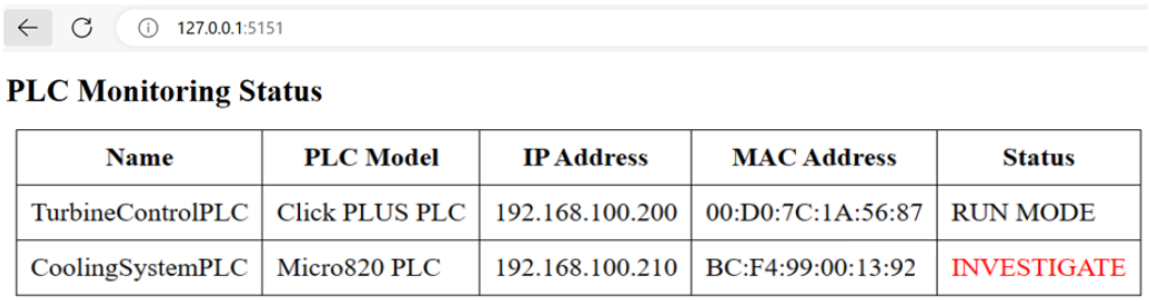


Figure 29. Updated Web Application for Monitoring PLC Operational Modes

An administrative web page was provided to allow for additional PLCs of known supported types to be added as well.

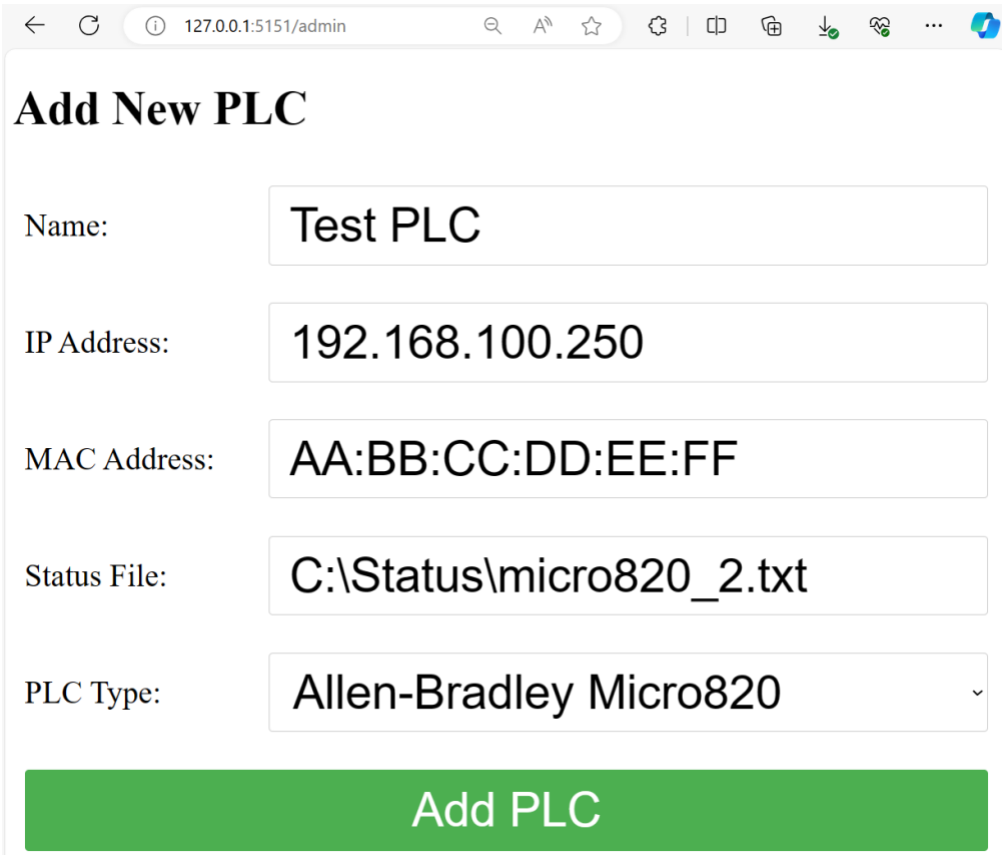


Figure 30. Administrative Page for Adding New PLCs for Monitoring

# 4. Recommendations and Implications

During the research process, several opportunities for improving the cyber security posture of ICS/OT environments were identified. Many of these opportunities are overlooked in today's ICS/OT networks. Additionally, the research lays the foundation for future work to continue elevating awareness around the need for additional security of PLCs.

## 4.1. Recommendations for Practice

Based on the conducted research, asset owners and operators should consider implementing the following recommendations as part of their ICS/OT cyber security management program:

- Operators must understand that each PLC type and its operational states are different; it is essential to be familiar with each and understand their meaning.
- All PLCs in the environment should be monitored for changes in operational state, most importantly, coming out of “Run” mode.
    - PLCs should be monitored for any code changes, whether this is the PLC programming code itself or the PLC's firmware.
- Implement a monitoring system to determine changes in the operational state (and, where possible, changes in programming and firmware).
- If a PLC is determined to be taken out of “Run” mode, verify if an authorized change is being made at that time. If an authorized change is not currently underway, investigate the cause of the PLC's operational mode change.
    - Investigating a change in operational mode could help identify an operational process issue or suspicious activity for further investigation.
- Follow the guidance in the Secure PLC Coding Practices: Top 20 List on further ways to help secure PLCs.

### 4.2. Implications for Future Research

The research here is only a start, limited to primarily a single PLC from one vendor. Future research should include, but not be limited to, the following:

- Ability to add other PLC types from different manufacturers for monitoring.
- Different form factors of PLCs, such as virtualized and cloud-based, should be included.
- The entire attack surface of each PLC should be thoroughly examined to help better understand how to protect these critical assets.
- Tools can be developed to help evaluate the security of these assets, helping operators proactively identify security vulnerabilities before an attacker does.

Further research could help provide additional security approaches for ensuring PLC cyber security while offensive tools for testing PLC security could be developed.

## 5. Conclusion

The security of PLCs can often be overlooked and misunderstood in Industrial Control Systems and Operational Technology environments. Not all ICS/OT environments tend to be concerned with the cyber security of the PLCs in their environment, preferring to focus on securing higher levels of the network. With that said, there is a general understanding that a PLC can be kept secure if it remains in “Run” mode, preventing remote attackers from being able to update the PLC programming and firmware. Unfortunately, that blanket statement cannot be applied to all brands and models of PLCs. This research demonstrates that programming changes to a PLC can still be made to certain PLCs even when in “Run” mode. While monitoring for changes to the PLC, such as when it comes out of “Run” mode, is still an essential aspect of any OT cyber security monitoring program, it is not the silver bullet many might make it out to be. Operators must be very familiar with how each of their PLCs operate, including the different operational modes for each and any associated security risks. Such operational

risks could impact not only the uptime of the facility but also, most importantly, the safety of onsite personnel, the general public, and the surrounding environment.

# References

AutomationDirect.com. (2023). CLICK PLUS PLC user manual (C0-USER-M) (6th ed., Rev. M) [PDF]. https://www.automationdirect.com/

Ackerman, P. (2021). *Industrial Cybersecurity* (2nd ed.). Packt Publishing Ltd.

Bodungen, C. E., Singer, B. L., Shbeeb, A., Hilt, S., & Wilhoit, K. (2017). Hacking Exposed Industrial Control Systems: ICS and SCADA Security Secrets & Solutions (1st ed., pp. 149-153). McGraw-Hill Education.

Brooks, C. J., & Craig, P. A., Jr. (2022). Practical Industrial Cybersecurity: ICS, Industry 4.0, and IIoT. Wiley.

Dragos. (2023). 2022 ICS/OT Cybersecurity Year in Review. https://www.dragos.com/year-in-review/

Greenberg, A. (2018). *Sandworm: A New Era of Cyberwar and the Hunt for the Kremlin's Most Dangerous Hackers* (3rd ed.). Anchor.

Hoffman, M., & Cedillo, G. (2021, December 9). Detecting PLC switch position changes through the network. Dragos. https://www.dragos.com/blog/detecting-plc-switch-position-changes-through-the-network/

Hoffman, M., & Cedillo, G. (2023, November 14). The value of PLC key switch monitoring is to keep critical systems more secure. Dragos. https://www.dragos.com/blog/industry-news/value-of-plc-key-switch-monitoring/

Anonymous. (2021, June 15). Top 20 secure PLC coding practices. https://www.plc-security.com/content/Top_20_Secure_PLC_Coding_Practices_V1.0.pdf

Mustard, S. (2022). Industrial Cybersecurity Case Studies and Best Practices (1st ed.). International Society of Automation (ISA).

Mike Holcomb, mike@mikeholcomb.com

The MITRE Corporation. (2023, November 22). Techniques - ICS | MITRE ATT&CK. Retrieved December 2, 2023, from https://attack.mitre.org/techniques/ics/

Dragos. (n.d.). TRISIS malware: Analysis of safety system targeted malware. Retrieved December 9, 2023, from https://www.dragos.com/wp-content/uploads/TRISIS-01.pdf

Rockwell Automation. (2023). Micro820 programmable controllers user manual. https://literature.rockwellautomation.com/idc/groups/literature/documents/um/2080-um005_-en-e.pdf