TryHackMe Advent of Cyber 2025 Day 19 Challenge Report

ICS/SCADA Systems & Modbus Protocol Security

1. Executive Summary

This report documents the completion of Day 19 of the TryHackMe Advent of Cyber 2025 event. The challenge focused on Industrial Control Systems (ICS) security, SCADA architecture, PLC operations, and Modbus protocol vulnerabilities. Successfully investigated the compromised TBFC drone control system, identified malicious configuration changes via unauthenticated Modbus TCP access, navigated sophisticated trap mechanisms designed to prevent remediation, and safely restored Christmas deliveries using Python's pymodbus library. Retrieved both challenge flags demonstrating practical understanding of industrial automation security, protocol analysis, and safe incident response in critical infrastructure environments.

2. Understanding SCADA Systems

2.1 What is SCADA?

SCADA (Supervisory Control and Data Acquisition) systems are the command centers of industrial operations. They act as the critical bridge between human operators and automated machinery performing physical work. SCADA functions as the nervous system of modern factories, warehouses, power plants, and water treatment facilities-continuously sensing conditions, processing information, and sending commands to make industrial processes happen efficiently and safely.

In TBFC's case, the SCADA system oversees the entire drone delivery operation. Without it, operators would have no way to monitor hundreds of drones simultaneously, manage inventory levels across multiple warehouses, track package routing, or ensure deliveries reach correct destinations. It serves as the invisible orchestrator of Christmas logistics, coordinating thousands of automated decisions every minute.

2.2 SCADA System Components

A typical SCADA system consists of four essential components working in coordination:

Sensors & Actuators:

Sensors serve as the system's eyes, continuously measuring real-world conditions including temperature, pressure, position, weight, speed, and flow rate. Actuators function as the system's hands, performing physical actions-motors turn conveyor belts, valves open to control fluid flow, robotic arms move packages, and mechanical switches activate equipment. In TBFC's warehouse, sensors detect when packages arrive on conveyor belts (weight sensors), identify package types (barcode scanners), and verify drone loading positions (proximity sensors). Actuators control the robotic arms that pick packages, position them correctly, and load them onto waiting drones.

PLCs (Programmable Logic Controllers):

PLCs are the intelligent brains executing automation logic. They continuously read sensor data, make decisions based on programmed rules and conditions, and send precise commands to actuators. A PLC might implement logic such as: 'IF package weight equals 2kg AND destination equals Zone 5 AND Drone 7 is available THEN activate robotic arm #3 to load package onto Drone 7.' These decisions happen in milliseconds, enabling real-time process control. PLCs are purpose-built industrial computers designed for extreme reliability, operating 24/7 for years without failure in harsh environments with dust, vibration, temperature extremes, and electromagnetic interference.

Monitoring Systems:

Visual interfaces provide human operators with real-time observation capabilities through CCTV cameras, graphical dashboards, alarm panels, and status displays. TBFC's warehouse security cameras (accessible on port 80) show live footage of the packaging floor, enabling operators to watch what the automation is doing, verify system behavior matches expectations, and identify anomalies requiring intervention. These monitoring systems provide immediate visual feedback-you can literally watch packages moving on conveyors, robotic arms operating, and drones launching for delivery.

Historians:

Historian databases store every operational data point for later analysis and compliance. Every package loaded gets recorded with timestamp, type, weight, destination, and assigned drone. Every system configuration change, operator login, alarm activation, and process deviation gets logged. This historical data serves multiple purposes: identifying performance patterns, troubleshooting operational problems, optimizing efficiency, ensuring regulatory compliance, and-critically for incident response scenarios like ours-reconstructing exactly what an attacker did, when they did it, and what systems they compromised.

2.3 SCADA in TBFC Drone Delivery

The compromised TBFC SCADA system manages several critical operational functions:

- **Package Type Selection:** System determines whether to load Christmas gifts, chocolate eggs, or Easter baskets onto each drone. This selection is controlled by a simple numeric value (stored in Holding Register 0) that determines which conveyor belt activates and which inventory gets accessed.
- **Delivery Zone Routing:** Each package must reach the correct neighborhood. Zones 1-9 represent different districts of Wareville with specific GPS coordinates. Zone 10 is reserved for emergency disposal-packages damaged during processing get routed to ocean disposal (a failsafe mechanism, but also a perfect target for sabotage).
- **Visual Monitoring:** CCTV camera feeds provide real-time observation of warehouse operations. Operators can view which items are being loaded, verify system behavior matches configuration, and identify anomalies. This visual layer becomes crucial during incident response-it provides immediate confirmation of whether remediation efforts are working.
- **Inventory Verification:** Before loading a package, the system can check whether the requested item actually exists in physical stock. When this verification is enabled (Coil 10 = True), the system queries the inventory database before executing commands. When disabled, the system blindly follows commands-even if those commands reference non-existent items or are malicious.

- **System Protection Mechanisms:** Security features designed to prevent unauthorized changes and detect tampering. When enabled (Coil 11 = True), these protections monitor for suspicious modifications and can trigger defensive responses. Unfortunately, King Malhare weaponized these very protections as part of his trap-a clever attacker technique of turning security features against defenders.
- **Audit Logging:** Every configuration change, operator login, system modification, and operational event should be recorded with precise timestamps for accountability and forensic investigation. Attackers routinely disable logging (Coil 13 = False) to cover their tracks-and that's precisely what happened in this incident.

2.4 Why SCADA Systems Are Targeted

Industrial control systems have become increasingly attractive targets for cybercriminals, hacktivists, and nation-state actors. Understanding why helps defenders prioritize security:

- **Legacy Software with Known Vulnerabilities:** Many SCADA systems were installed decades ago when cybersecurity wasn't a primary concern. They continue running ancient operating systems (Windows XP, Windows Server 2003) and software versions with publicly documented exploits. Updating these systems is difficult because: (1) vendors may no longer support old hardware, (2) industrial processes require extensive testing before changes, (3) downtime for updates costs millions in lost production.
- **Default Credentials Never Changed:** System administrators prioritize operational uptime over security hardening. Default usernames and passwords (admin/admin, root/root) persist because the mentality is 'if it works, don't touch it.' Industrial environments value stability-any change risks production disruptions.
- **Designed for Reliability, Not Security:** Most SCADA systems were engineered when networks were physically isolated (air-gapped). Security through obscurity was the model-if attackers can't reach the network, they can't compromise systems. Authentication, encryption, and access controls were afterthoughts at best. Modbus protocol (created 1979) has zero built-in security.
- **Control Physical Processes:** Unlike attacking websites (data theft) or compromising servers (service disruption), SCADA attacks have real-world physical consequences. Attackers can cause power blackouts affecting millions, contaminate water supplies threatening public health, disrupt manufacturing causing economic damage, or-in our scenario-sabotage Christmas deliveries causing disappointment to children worldwide.
- **Connected to Corporate Networks:** The myth of air-gapped industrial systems is largely fiction. Most SCADA environments connect to business networks for: remote monitoring by engineers, data integration with ERP systems, automated reporting to management, and vendor support access. These connections provide attackers with entry points-compromise the corporate network, pivot to SCADA network.
- **Protocols Lack Authentication:** Industrial protocols like Modbus, DNP3, and BACnet were designed for trusted environments. They assume anyone on the network is authorized. Modbus TCP has no authentication mechanism-anyone who can reach port 502 can read values, write values, and modify system configuration without proving identity.

In early 2024, the first ICS/OT-specific malware framework called FrostyGoop was discovered in the wild. This malware can directly interface with industrial control systems via the Modbus TCP protocol, enabling arbitrary reads and writes to device registers over TCP port 502. King Malhare employed these exact tactics-not to cause power blackouts, but to sabotage Christmas deliveries by directly manipulating control logic through unauthenticated Modbus access.

3. Understanding PLCs

3.1 What is a PLC?

A PLC (Programmable Logic Controller) is an industrial computer purpose-built to control machinery and processes in real-world environments. Unlike consumer computers (laptops, smartphones, servers), PLCs are specialized devices engineered for extreme reliability, harsh conditions, and real-time control. They're designed to operate 24/7/365 for decades without rebooting, withstand temperature extremes (-40°C to 70°C), survive constant vibration and electromagnetic interference, execute control logic in milliseconds, and interface directly with industrial electrical systems (24V DC, 120V AC, 4-20mA current loops).

4. Modbus Protocol Deep Dive

4.1 What is Modbus?

Modbus is the communication protocol industrial devices use to exchange data. Created in 1979 by Modicon (now Schneider Electric) for factory automation, it's one of the oldest and most widely deployed industrial protocols globally. Its longevity isn't due to sophisticated features-quite the opposite. Modbus succeeded because it's extremely simple, highly reliable, works with virtually any device, and easy to implement and debug.

Think of Modbus as a basic request-response conversation: Client asks 'PLC, what's the current value of register 0?' and Server responds 'Register 0 currently holds value 1.' This simplicity makes Modbus easy to troubleshoot but also means security was never a consideration. There's no authentication (anyone can connect), no encryption (all data transmitted plaintext), no authorization (no permission system), and no integrity verification (beyond basic checksums for transmission errors).

4.2 Modbus Data Types

Modbus organizes data into four distinct types, each serving specific purposes in industrial automation:

- **Coils (Outputs):** Read/write boolean values (True/False, On/Off). Used for digital outputs controlling motors, valves, lights, alarms. Example: Coil 10 = Inventory Verification Enabled.
- **Discrete Inputs:** Read-only boolean values. Represent physical switch states, sensor status. Example: Emergency stop button pressed, door interlock closed.
- **Holding Registers:** Read/write 16-bit numeric values (0-65535). Store configuration settings, setpoints, control parameters. Example: HR0 = Package Type Selection.
- **Input Registers:** Read-only 16-bit numeric values. Represent sensor measurements, analog inputs. Example: Temperature sensor reading, pressure gauge value.

The critical distinction: Coils and Holding Registers are writable-attackers can change values to control system behavior. Discrete Inputs and Input Registers are read-only-they reflect sensor measurements you observe but cannot directly modify (though attackers might manipulate physical sensors).

4.3 TBFC System Register Map

The crumpled note found in the warehouse documented the complete Modbus address map:

Holding Registers:

- HR0: Package Type Selection (0=Christmas Gifts, 1=Chocolate Eggs, 2=Easter Baskets)
- HR1: Delivery Zone (1-9 for normal zones, 10 for emergency ocean disposal)
- HR4: System Signature/Version (Default=100, Compromised=666 Eggsploit signature)

Coils (Boolean Control Flags):

- C10: Inventory Verification (True=Check actual stock, False=Blind operation)
- C11: Protection/Override (True=Changes locked/monitored, False=Normal operation)
- C12: Emergency Dump Protocol (True=DUMP ALL INVENTORY, False=Normal)
- C13: Audit Logging (True=All changes logged, False=No logging)
- C14: Christmas Restored Flag (Auto-set when system correct)
- C15: Self-Destruct Status (Auto-armed on unauthorized configuration change)

4.4 Modbus TCP vs Serial

Originally, Modbus operated over serial connections (RS-232, RS-485 cables). Devices physically connected in networks, and this physical isolation provided security-attackers needed physical access to wiring. Modern systems use Modbus TCP, encapsulating Modbus protocol inside standard TCP/IP packets on port 502. This brings benefits (remote monitoring, easier integration) but exposes historically isolated systems to network attacks. King Malhare exploited this-TBFC's Modbus TCP port was network-accessible without authentication.

5. Investigation Procedure

5.1 Port Scanning

```
nmap -sV -p 22,80,502 10.66.151.233
```

Results:

- Port 22: SSH (OpenSSH 9.6p1)
- Port 80: HTTP (Werkzeug - CCTV camera feed)
- **Port 502: Modbus TCP - CRITICAL VULNERABILITY**

5.2 Visual Confirmation

Accessed http://10.66.151.233 CCTV feed. Visual observation confirmed: pastel-colored chocolate eggs being sorted, Easter-themed packaging on assembly lines, delivery drones loading eggs instead of Christmas presents, status display showing 'Compromised'. This confirmed logic manipulation attack-system working perfectly, just delivering wrong items.

5.3 Modbus Investigation with Python

Used Python pymodbus library to interrogate Modbus server directly:

```
from pymodbus.client import ModbusTcpClient client = ModbusTcpClient('10.66.151.233',
port=502) client.connect()
```

Critical Findings:

- **HR0 = 1:** Package type set to Chocolate Eggs (should be 0 for Christmas)
- **HR1 = 5:** Delivery zone normal (not ocean dump)
- **HR4 = 666:** EGGSPLOIT signature - attacker calling card
- **C10 = False:** Inventory verification DISABLED
- **C11 = True:** Protection ACTIVE - TRAP MECHANISM ARMED
- **C12 = False:** Emergency dump not yet triggered
- **C13 = False:** Audit logging DISABLED - covering tracks
- **C14 = False:** Christmas not restored
- **C15 = False:** Self-destruct not armed YET

6. Trap Mechanism Analysis

6.1 Understanding the Trap

Critical Discovery from Maintenance Note:

"Never change HR0 whilst C11=True! Will trigger countdown!"

Trap Logic Chain:

1. C11 (Protection) is enabled, actively monitoring for configuration changes
2. If HR0 is modified while C11=True, trap triggers immediately
3. C15 (Self-Destruct) arms automatically
4. 30-second countdown begins (irrevocable)
5. After 30 seconds, C12 (Emergency Dump) activates
6. HR1 changes to 10 (ocean disposal zone)
7. ALL remaining inventory dumps to ocean - total sabotage

This sophisticated trap mechanism demonstrates advanced attacker tradecraft-weaponizing the system's own security protections against defenders. The maintenance technician who wrote the note likely discovered this trap during investigation but was interrupted before remediation.

7. Safe Remediation Procedure

7.1 Correct Sequence

Step 1: Disable Protection (C11) FIRST

```
client.write_coil(11, False, slave=1)
```

This disarms the trap mechanism before making any configuration changes.

Step 2: Change Package Type to Christmas

```
client.write_register(0, 0, slave=1)
```

HR0 = 0 sets package type to Christmas presents. Safe now that C11 is disabled.

Step 3: Enable Inventory Verification

```
client.write_coil(10, True, slave=1)
```

Step 4: Enable Audit Logging

```
client.write_coil(13, True, slave=1)
```

Step 5: Verification

Confirmed C14 (Christmas Restored) auto-set to True, C15 (Self-Destruct) remained False, CCTV feed showed Christmas presents being loaded.

8. Challenge Answers

Question 1: What port is commonly used by Modbus TCP?

Answer: 502

Question 2: Flag after restoration?

Flag: THM{eGgMas0V3r}

9. Key Skills Developed

- SCADA system architecture understanding
- PLC operational principles
- Modbus protocol analysis
- Python pymodbus library usage
- Industrial control system compromise investigation
- Trap mechanism identification and navigation
- Safe incident response in critical infrastructure
- ICS security vulnerability assessment

10. Conclusion

Day 19 of the TryHackMe Advent of Cyber 2025 provided comprehensive hands-on training in Industrial Control System security, demonstrating real-world SCADA vulnerabilities and attack techniques. Successfully investigated a sophisticated ICS compromise where the attacker leveraged unauthenticated Modbus TCP access to manipulate PLC configuration, implemented trap mechanisms to prevent remediation, and weaponized system security features against defenders.

The challenge highlighted critical ICS security lessons: Modbus protocol's complete lack of authentication makes it fundamentally vulnerable to network-based attacks, legacy industrial systems often prioritize operational uptime over security hardening, physical process control through cyber means can have real-world consequences, and understanding attacker techniques is essential before attempting remediation in critical systems. King Malhare's Eggsploit framework (signature 666) demonstrated how attackers can turn industrial automation against its intended purpose-not just breaking systems, but making them work perfectly while executing malicious logic. The maintenance technician's warning about the trap mechanism proved essential-attempting remediation without understanding the protection logic would have triggered the self-destruct countdown, making the situation catastrophically worse.

Challenge Status: COMPLETED ✓ - Christmas Restored!