
Countdown to
Industrial Extinction

Disclaimer

- ❑ The information provided in this course is for informational purposes only.
- ❑ All knowledge obtained must be used for legitimate, authorized purposes.
- ❑ The views written and discussed in this course are those of the instructor and not any of their employers or other affiliated organizations.

About Mike Holcomb

- ❑ Fellow for Cyber Security at Fluor
- ❑ ICS/OT Cyber Security Global Lead
- ❑ Founder of Upstate SC ISSA and BSides Greenville
- ❑ Wrote and taught all six cyber security courses for Greenville Technical College's cyber degree program
- ❑ CISSP, GRID, GICSP, ISA 62443, GPEN, GCIH, etc.
- ❑ Master's degree in ICS/OT cyber security from the SANS Technology Institute

About Mike Holcomb

About Mike Holcomb

Mike Holcomb (He/Him)

Helping You Secure ICS/OT | Fellow, ICS/OT Cybersecurity Global Lead

Top Cybersecurity Voice

Greenville, South Carolina, United States · Contact info

27,183 followers · 500+ connections

Fluor

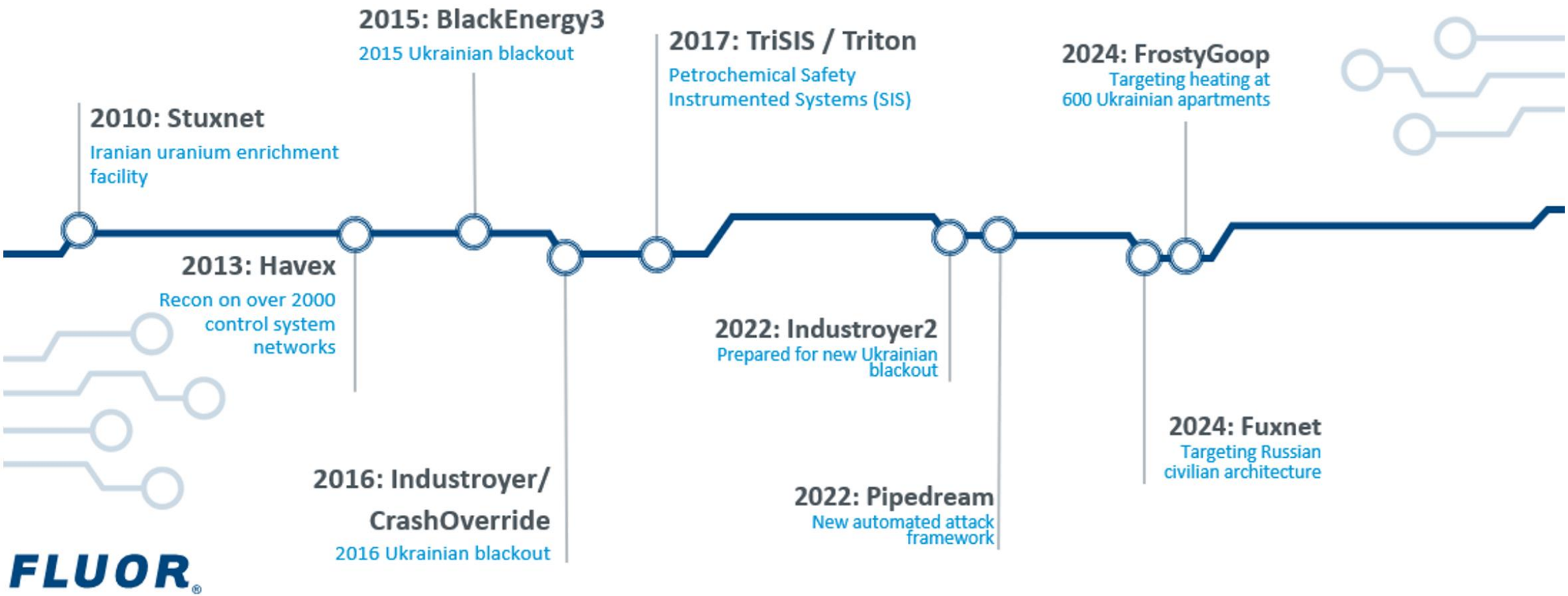
SANS Technology Institute

This is Megadeth!

What Does a Hacked ICS/OT System Look Like?

Source: reddit.com/r/PLC

ICS-Specific Malware



FLUOR®

Annotated ICS/OT Cyber Security History

- ❑ 2003: SQL Slammer brings down the Davis-Besse plant
- ❑ 2005: The movie ‘Hackers’ was released
- ❑ 2009: Conficker infects power plants in the U.S.
- ❑ 2010: The ICS/OT cyber security world awakens

Stuxnet

Annotated ICS/OT Cyber Security History

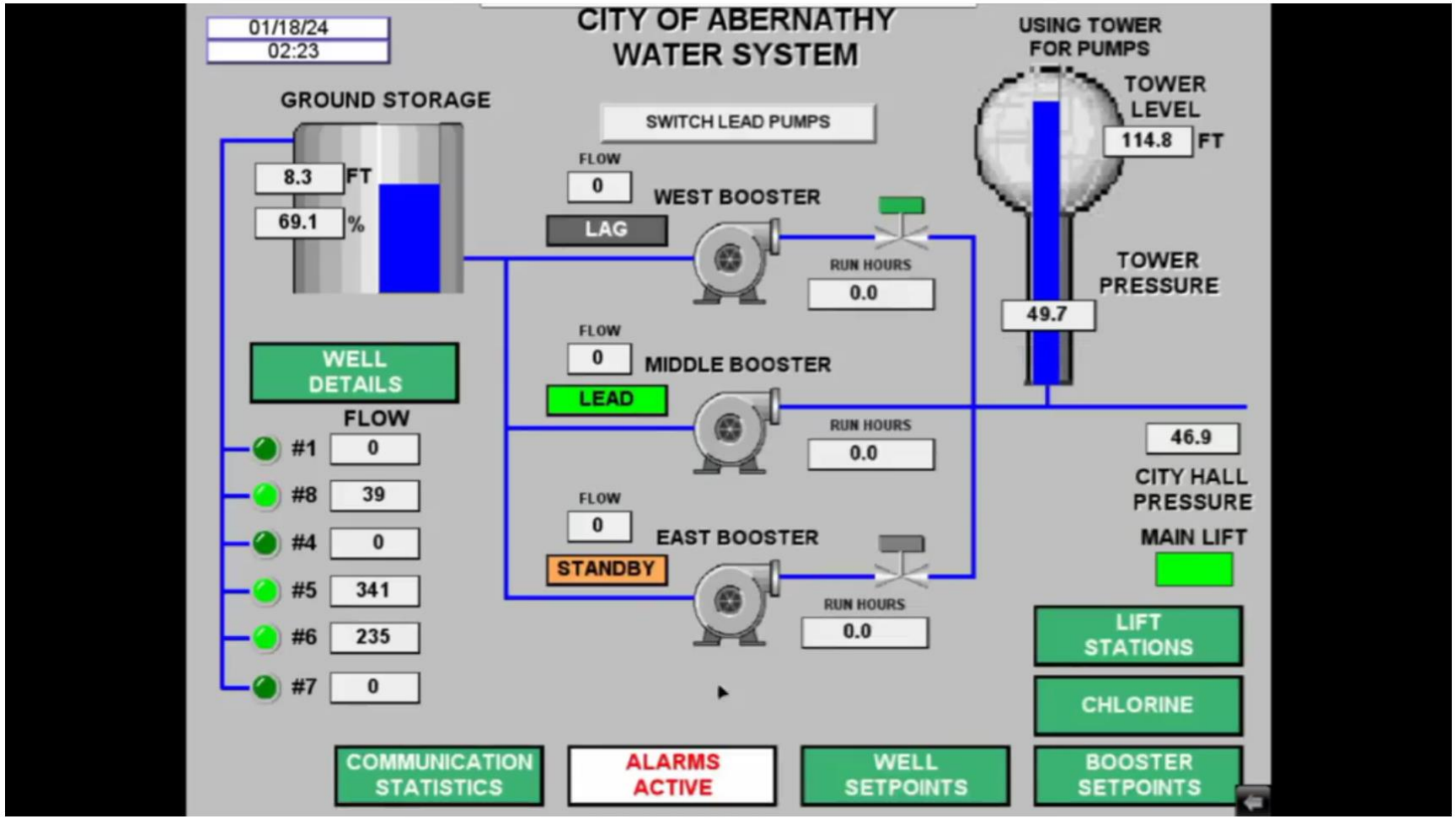
- ❑ 2003: SQL Slammer brings down the Davis-Besse plant
- ❑ 2005: The movie ‘Hackers’ was released
- ❑ 2009: Conficker infects power plants in the U.S.
- ❑ 2010: Stuxnet
- ❑ 2015: Ukrainian power grid taken offline
- ❑ 2016: Ukrainian power grid taken offline (yes, again)
- ❑ 2017: Attackers compromise Safety Instrumented Systems (SIS)
- ❑ 2021: The ICS/OT world changes again

The ICS/OT World Changes Again

What Has Happened Since Then?

- Coordinated attack against 22 utility providers in Denmark
- Mandiant announce 2022 Ukrainian power outage result of Russian cyber attack
- Volt Typhoon attackers targeting critical infrastructure
  - Originally discovered in Guam; discovered within the US
- Ports paralyzed by ransomware attacks
- Fuxnet used to disrupt 28,000 safety and control sensors in Russia
- Hacktivists, and groups pretending to be hacktivists, continue to target various utilities

The Cyber Army of Russia Reborn (CARR)



Source: wired.com/story/cyber-army-of-russia-reborn-sandworm-us-cyberattacks/

FrostyGoop

- 9th known malware to specifically target ICS/OT assets
- Targets the most used ICS/OT protocol - Modbus TCP
- First malware to have an ICS/OT impact using Modbus TCP
- Targeted an energy company to impact heating to 600 apartment buildings in the Ukraine in sub-zero temperatures
- Heating was down for two days
- First malware to have an ICS/OT impact using Modbus TCP

FrostyGoop

- 9th known malware to specifically target ICS/OT assets
- Targets the most used ICS/OT protocol - Modbus TCP
- First malware to have an ICS/OT impact using Modbus TCP
- Targeted an energy company **to impact heating to 600 apartment buildings in the Ukraine in sub-zero temperatures**
- Heating was down for two days
- First malware to have an ICS/OT impact using Modbus TCP

FrostyGoop

- 9th known malware to specifically target ICS/OT assets
- Targets the most used ICS/OT protocol - Modbus TCP
- First malware to have an ICS/OT impact using Modbus TCP
- Targeted an energy company **to impact heating to 600 apartment buildings in the Ukraine in sub-zero temperatures**
- **Heating was down for two days**
- First malware to have an ICS/OT impact using Modbus TCP

And the rate is accelerating.

Some are still ignoring the signs.

Many are waiting for someone to die before taking proactive steps.

What Can We Do?

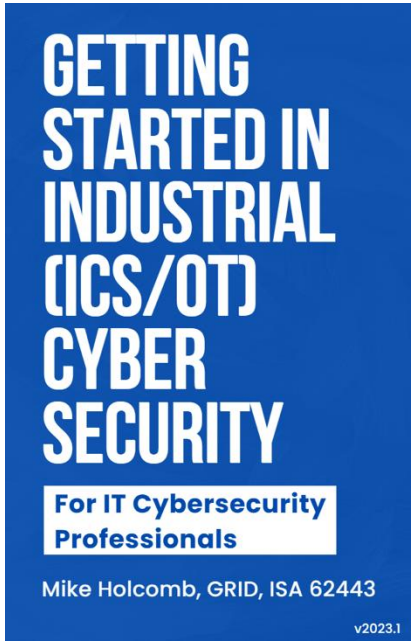
- ❑ Education
- ❑ IT & OT Collaboration
- ❑ OT Cyber Security Controls

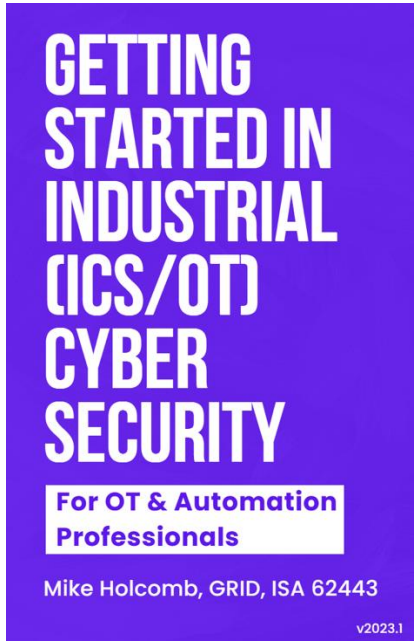
Owners and Operators ARE Targets!

- ❑ From the IT back office/enterprise network
- ❑ Physically brought in by “transitory cyber assets”
- ❑ Control systems exposed directly to the Internet
- ❑ Via remote access capabilities
- ❑ Malicious insiders

Education Resources

- CISA ICS/OT Cyber Security Courses
- “Getting Started with Industrial (ICS/OT) Cyber Security”
  - Free eBook and YouTube course





Education Resources

- CISA ICS/OT Cyber Security Courses
- “Getting Started with Industrial (ICS/OT) Cyber Security”
  - Free eBooks and YouTube course

IT & OT Collaboration?

"ICS cyber security is just like IT cyber security."

IT & OT Collaboration?

"ICS cyber security is NOT IT cyber security."

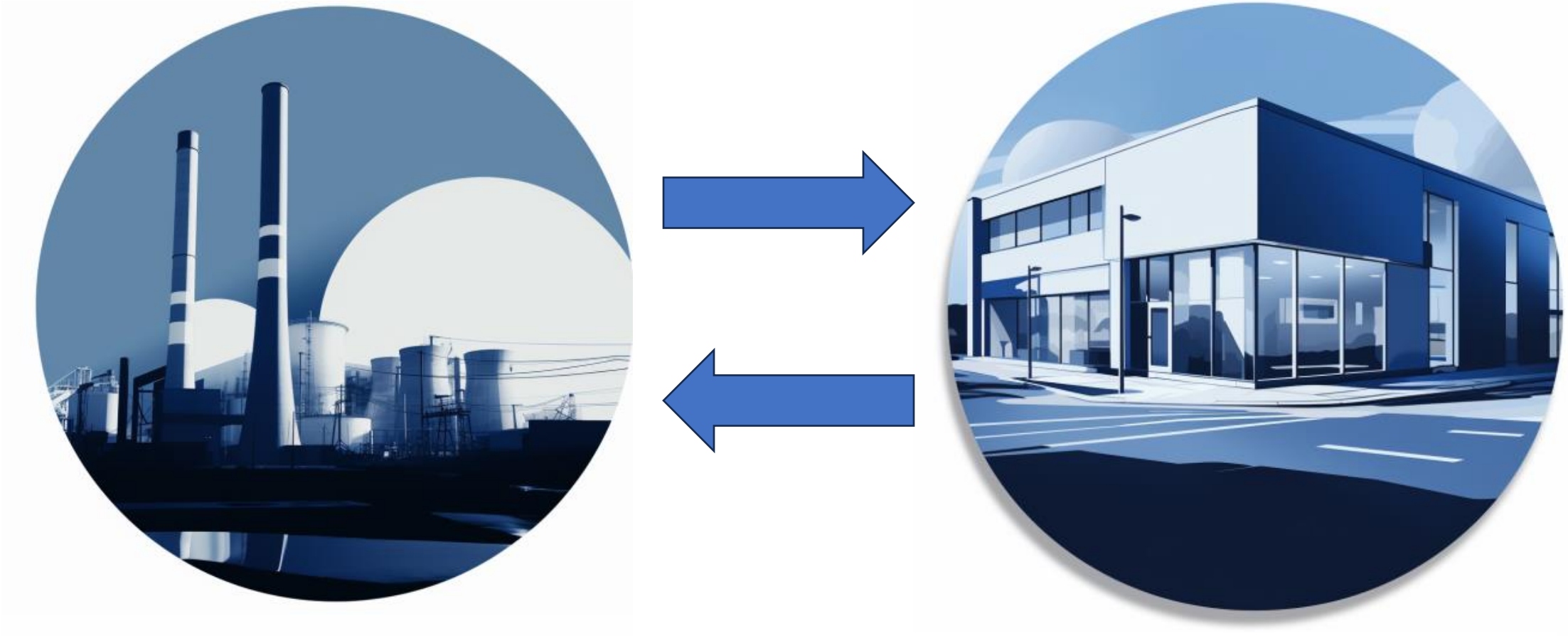
IT & OT Collaboration?

"ICS cyber security is **more like** IT cyber security **than it is different**."

The OT Side of the House

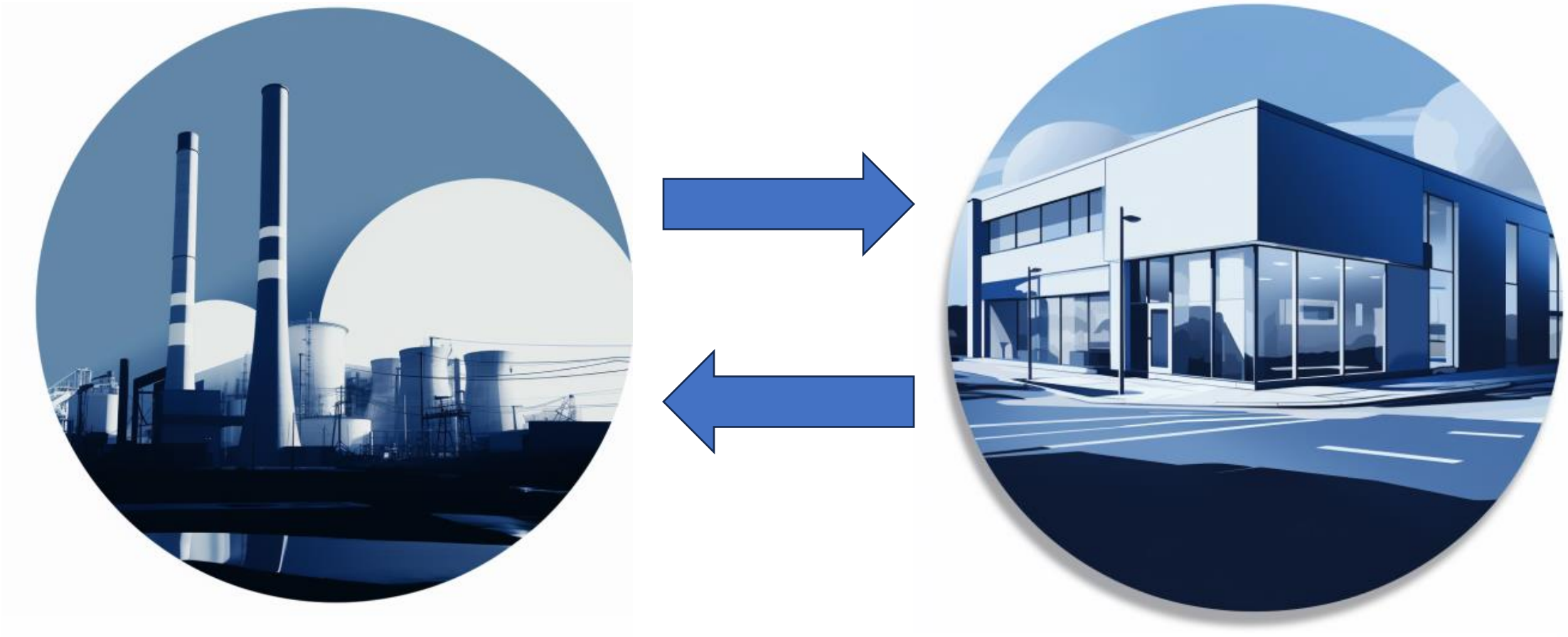
The IT Side of the House

OT & IT Communication



Can Become OT vs IT

We All Live in the Same House



Demonstrate the Need for Security Controls

How simple can it be to impact an ICS/OT network?

Do You Want to Build a Snowman?

Let’s build our own FrostyGoop.

Leveraging Vendor Documentation

- ❑ Dragos discovered and reported on FrostyGoop
- ❑ Related webinars can be a great source of knowledge and inspiration
- ❑ Download the full report at https://www.dragos.com/resources/reports/intelligence-brief-impact-of-frostygoop-modbus-malware-on-connected-ot-systems/

FrostyGoop Main Capabilities

FrostyGoop ICS Malware Capabilities

- Accepts optional command line execution arguments.
- Uses separate configuration files to specify target IP addresses and Modbus commands.
- Communicates with ICS devices via Modbus TCP protocol.
- Sends Modbus commands to read or modify data on ICS devices.
- Logs output to a console or JSON file.

Source: Impact of FrostyGoop ICS Malware on Connected OT Systems
(Mark Graham, Carolyn Ahlers, Kyle O-Meara of Dragos)

FrostyGoop – Command Line Arguments

Optional Command Line Execution Arguments

FrostyGoop checks if the executable is running with any required command line arguments. The binaries exit execution if no command line arguments are provided. The specific arguments vary by sample, but functionality remains the same. Information required to initiate a TCP connection and send Modbus commands to a victim ICS device can be specified as command-line arguments or contained within a separate JSON configuration file. Arguments accepted by FrostyGoop would include data such as:

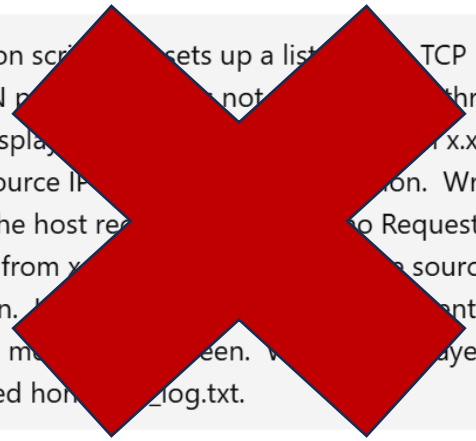
- IP addresses specifying the target device to communicate with
- A “mode” option that correlates to a Modbus command to execute on the ICS device (Read Holding Registers, Write to Single Holding Register, Write to Multiple Holding Registers)
- A Modbus register address on the target ICS device to send Modbus commands to
- A JSON configuration file name; there are two different configuration files accepted by FrostyGoop
  - A configuration file containing victim device information such as IP address, Modbus commands, and Modbus register addresses
  - A configuration file containing a specific time to begin Modbus TCP communications with the victim device and various lengths to delay the execution of Modbus commands.
- Specify a file name to save logging output

Source: Impact of FrostyGoop ICS Malware on Connected OT Systems (Mark Graham, Carolyn Ahlers, Kyle O-Meara of Dragos)

Using ChatGPT to Create an Offensive Tool

Prompt:

Create a Python scr… sets up a list… TCP port 502. If it receives a SYN p… not… three-way handshake, displa… x.x.x.x ** where x.x.x.x is the source IP… on. Write a message to the screen if the host re… o Request packet. ** ICMP Echo Request from x… source IP address of the connection. … nt happens when displaying the m… een. … yed output to a local file named hon… log.txt.



NO

Prompt:

Create a Python script that listens on TCP 502.

Update the script to display a message to the screen when the listening host receives an ICMP Echo Request.

Update the script to detect when a SYN packet is received but the three-way handshake is not completed. Display a message that reads ** POSSIBLE PORT SCAN from x.x.x.x **.

Update the script to include the date & time of when the event occurred.

Write all displayed output to a local file named honeypot_log.txt.

YES ✓

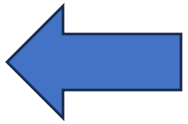
Starting to Build Our Own FrostyGoop

Prompt:

> Write a Python script named snowcrash.py. The user must supply an IP address at the command line.

Starting to Build Our Own FrostyGoop

Here's a simple template for a Python script named `snowcrash.py` that requires the user to supply an IP address as a command-line argument. This script includes basic input validation and a placeholder function for whatever operations you plan to perform with the IP address:



python Copy code

```python
import sys
import re

def validate_ip(ip_address):
    # Regular expression to validate an IP address
    ip_pattern = re.compile(r'^(\d{1,3}\.){3}\d{1,3}$')

    if ip_pattern.match(ip_address):
        # Split the IP address into its parts and check if each part is between 0 and 255
        parts = ip_address.split('.')
        if all(0 <= int(part) <= 255 for part in parts):
            return True
    return False

def perform_action(ip_address):
    # Placeholder function to perform actions with the IP address
    print(f"Performing action on {ip_address}")
```

Running Snowcrash for the First Time

```
c:\Scripts\Snowcrash>snowcrash.py
Usage: python snowcrash.py <IP_ADDRESS>
```

```
c:\Scripts\Snowcrash>snowcrash.py 192.168.1.200
Performing action on 192.168.1.200
```

Building Snowcrash Brick by Brick

Prompt:

Connect to the specified IP address using Modbus over TCP 502. Display a message on the screen to indicate that the destination host has Modbus available on TCP 502.

Building Snowcrash Brick by Brick

Building Snowcrash Brick by Brick (cont.)

```
c:\Scripts\Snowcrash>snowcrash.py 192.168.100.200

Modbus service is available on 192.168.100.200:502
```

Building Snowcrash Brick by Brick (cont.)

Prompt:

If the Modbus service is available at the target IP, provide the user with a menu of the following options: 1. Read register(s), 2. Write register(s), 3. Exit. If the user selects '1. Read register(s)', ask the user which register to start at. Then ask how many registers to read.

ChatGPT Creating Menu Choices

```
c:\Scripts\Snowcrash>snowcrash.py 192.168.100.200

Modbus service is available on 192.168.100.200:502


Menu:
1. Read register(s)
2. Write register(s)
3. Exit

Enter your choice: 1
Enter the starting register address: 0
Enter the number of registers to read: 10
Register values: [0, 1616, 0, 1616, 0, 0, 0, 0, 0, 0]
```

ChatGPT Making Assumptions

```
Menu:
1. Read register(s)
2. Write register(s)
3. Exit

Enter your choice: 2
Enter the register address to write to: 1
Enter the value to write: 0
Successfully wrote 0 to register 1.
```

ChatGPT Making Assumptions (cont.)

```
Menu:
1. Read register(s)
2. Write register(s)
3. Exit

Enter your choice: 1
Enter the starting register address: 0
Enter the number of registers to read: 10
Register values: [0, 0, 0, 1616, 0, 0, 0, 0, 0, 0]
```

Rebuilding The Actual FrostyGoop Command

Modbus.exe –ip=192.168.100.200 –mode=read –address=10 –count 10 –output=log.json

The user will specify if they want to issue a read register command with a -mode=read or if they want to issue a write register command with a -mode=write.

The user can use the -address=x argument to specify which register to start at.

The user can use the -count=x argument to specify how many registers to read.

The user can specify the -log=x.json argument where x.json is the name of a local file. Write all displayed output to this file.

The user can specify the -value=x argument where x is the new value to write to the specified register.

Rebuilding The Actual FrostyGoop Command

```
c:\Scripts\Snowcrash>snowcrash.py -ip=192.168.100.200  -mode=read -address=0 -count=10 -log=snow.json
Modbus service is available on 192.168.100.200:502
Register values: [0, 0, 0, 0, 72, 0, 0, 0, 0, 0]

c:\Scripts\Snowcrash>snowcrash.py -ip=192.168.100.200  -mode=write -value=5 -address=4 -log=snow.json
Modbus service i
Successfully wro

c:\Scripts\Snowc                                                              =10 -log=snow.json
Modbus service i
Register values:
```

snow.json

File Edit View


```
{"ip_address": "192.168.100.200", "command": "check_modbus_service", "result":
"Modbus service is available on 192.168.100.200:502", "success": true}
{"ip_address": "192.168.100.200", "command": "read_registers start=0 count=10",
"result": "Register values: [0, 0, 0, 0, 72, 0, 0, 0, 0, 0]", "success": true}
{"ip_address": "192.168.100.200", "command": "write_registers address=4 value=5",
"result": "Successfully wrote 5 to register 4.", "success": true}
{"ip_address": "192.168.100.200", "command": "read_registers start=0 count=10",
"result": "Register values: [0, 0, 0, 0, 5, 0, 0, 0, 0, 0]", "success": true}
```


Ln 2, Col 159 | 615 characters | 100% | Windows (CRLF) | UTF-8

One Last Really Cool Thing…

Prompt:

> Write a full descriptive help section to be displayed when the user has an issue.

Help is On The Way!

```
c:\Scripts\Snowcrash>snowcrash.py
Usage: python snowcrash.py -ip=<IP_ADDRESS> -mode=<read|write> -address=<START_ADDRESS> [-count=<NUMBER_OF_REGISTERS>]
 [-value=<VALUE_TO_WRITE>] [-log=<LOG_FILE>]

c:\Scripts\Snowcrash>snowcrash.py

Usage: python snowcrash.py -ip=<IP_ADDRESS> -mode=<read|write> -address=<START_ADDRESS>
                           [-count=<NUMBER_OF_REGISTERS>] [-value=<VALUE_TO_WRITE>] [-log=<LOG_FILE>]

Description:
  Snowcrash is a script for interacting with Modbus devices, allowing users to read or write
  to specific registers over TCP/IP. The script logs all operations to a specified log file.

Required Arguments:
  -ip=<IP_ADDRESS>           Specify the IP address of the Modbus device.
  -mode=<read|write>         Specify the operation mode:
                               - 'read' to read from Modbus registers.
                               - 'write' to write to a Modbus register.
  -address=<START_ADDRESS>   Specify the starting register address for read/write operations.

Optional Arguments:
  -count=<NUMBER_OF_REGISTERS>
                             Specify the number of registers to read. This option is required
                             when using the 'read' mode.
  -value=<VALUE_TO_WRITE>    Specify the value to write to a register. This option is required
                             when using the 'write' mode.
  -log=<LOG_FILE>            Specify the log file name to record all operations. Default is
                             'modbus_log.json'.
  -help, --help              Display this help message and exit.

Examples:
  To read 4 registers starting from address 1 on device 192.168.100.200:
    python snowcrash.py -ip=192.168.100.200 -mode=read -address=1 -count=4 -log=output_log.json

  To write the value 123 to register 1 on device 192.168.100.200:
    python snowcrash.py -ip=192.168.100.200 -mode=write -address=1 -value=123 -log=output_log.json
```

What Can We Do?

- ❑ Understand Which Assets are Running Modbus TCP
- ❑ Network Security Monitoring
- ❑ Network Segmentation
- ❑ SANS Critical Controls

But don’t stop there!

Critical Security Controls for ICS/OT

#1. Secure Network Architecture	#6. Conducting Risk Assessments	#11. Malware Defenses	#16. Access Control Management
#2. Asset Inventory Hardware/Software/Firmware	#7. Secure Remote Access	#12. Account Management	#17. Email & Browser Protection
#3. Incident Response Planning	#8. Employee Security Awareness	#13. Data Protection	#18. Application Software Security
#4. Backup & Recovery	#9. Network Security Monitoring	#14. Network Infrastructure Protection	#19. Service Provider Management
#5. Continuous Vulnerability Management	#10. IT / OT Partnership	#15. Technical Education for Team Members	#20. Penetration Testing

Critical Security Controls for ICS/OT

#1. Secure Network Architecture	#6. Conducting Risk Assessments	#11. Malware Defenses	#16. Access Control Management
#2. Asset Inventory Hardware/Software/Firmware	#7. Secure Remote Access	#12. Account Management	#17. Email & Browser Protection
#3. Incident Response Planning	#8. Employee Security Awareness	#13. Data Protection	#18. Application Software Security
#4. Backup & Recovery	#9. Network Security Monitoring	#14. Network Infrastructure Protection	#19. Service Provider Management
#5. Continuous Vulnerability Management	#10. IT / OT Partnership	#15. Technical Education for Team Members	#20. Penetration Testing

Thank You!

- For questions, comments or concerns, please don't hesitate to reach out!

Mike Holcomb
linkedin.com/in/mikeholcomb
youtube.com/@utilsec
github.com/utilsec
mikeholcomb.com
mike@mikeholcomb.com