



# Cold Reality: Impact of FrostyGoop Modbus ICS Malware Attacks on Connected OT Systems

Kyle O'Meara, Principal Adversary Hunter  
Carolyn Ahlers, Principal Malware Analyst  
Matt Pahl, Detection Technical Lead

# AGENDA

- 1 BACKGROUND

---
- 2 FROSTYGOOP ICS MODBUS MALWARE

---
- 3 INVESTIGATING THE INCIDENT

---
- 4 THREAT LANDSCAPE RESEARCH

---
- 5 RECOMMENDATIONS

---
- 6 THREAT INTEL > DETECTIONS

# FROSTYGOOP ICS MALWARE

**9<sup>th</sup>**  
known ICS  
malware

Dragos discovered FrostyGoop binaries in April 2024.

**1<sup>st</sup>**  
Modbus ICS  
malware that  
causes effects  
on ICS devices

FrostyGoop interacts directly with industrial control systems (ICS) using Modbus TCP over port 502.

# ATTACK ON CONNECTED ENERGY SYSTEMS

## What happened?

In January 2024, during sub-zero temperatures, a cyber attack disrupted the energy supply for central heating in more than 600 apartment buildings in Ukraine.

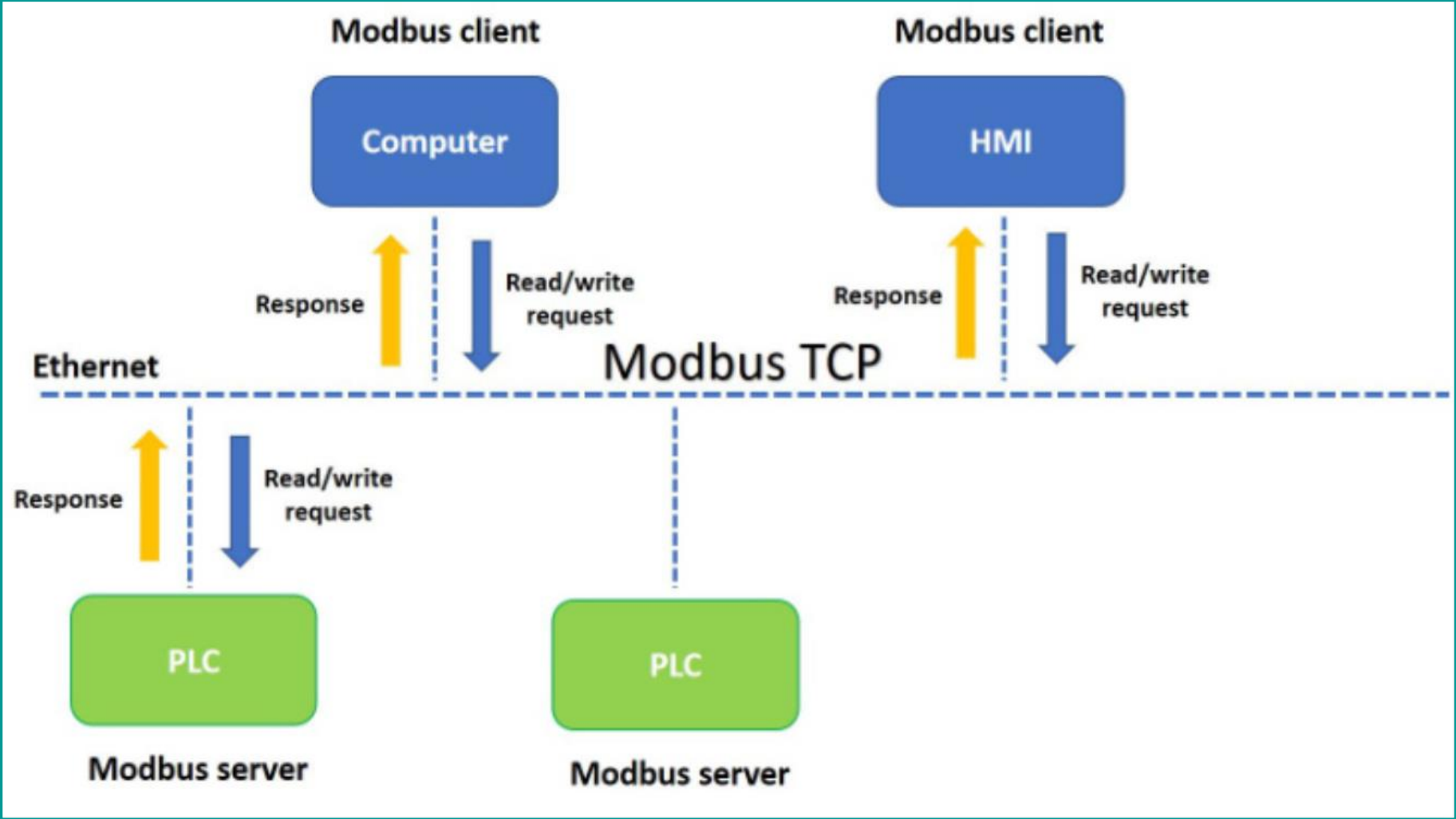


- The Cyber Security Situation Center (CSSC), part of the Security Service of Ukraine, shared details about this incident with Dragos.
- Dragos assessed that FrostyGoop was likely used to facilitate this attack.
- Dragos also assessed that before the attack, FrostyGoop was used to target controllers where TCP port 502 was internet-accessible.

# FROSTYGOOP MALWARE CAPABILITIES

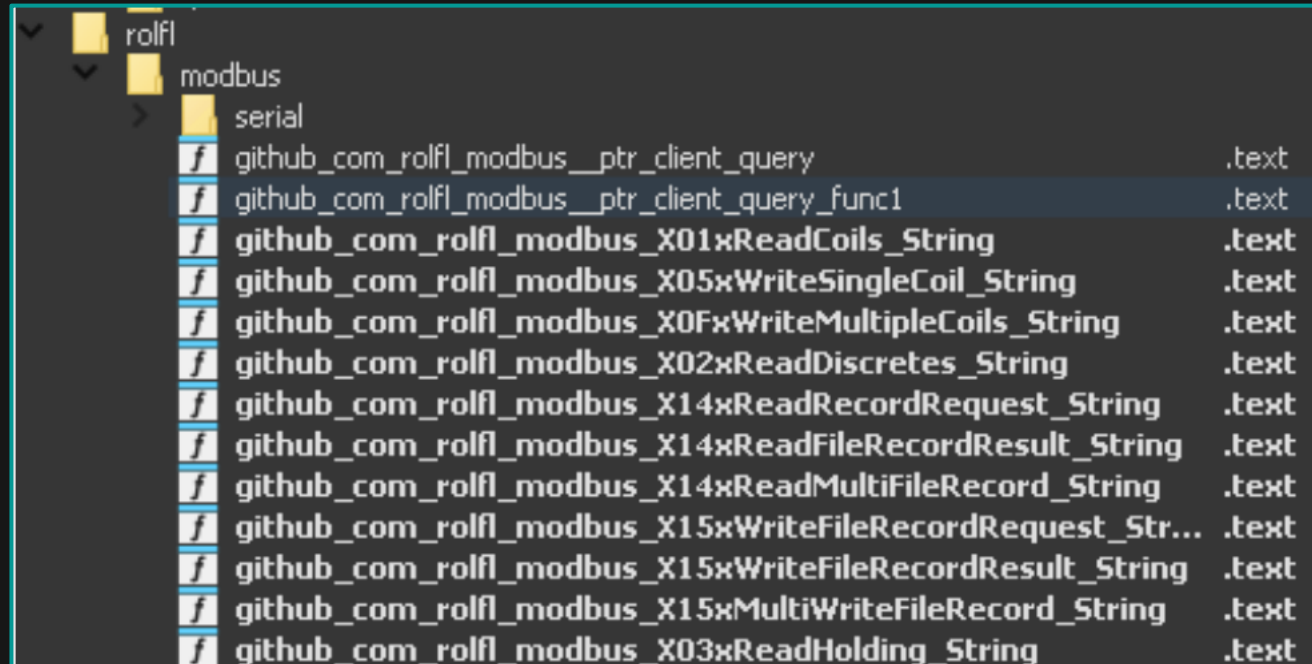
- Accept optional command line execution arguments
- Use a configuration file to specify target IP addresses and Modbus commands
- Communicate with ICS devices via Modbus TCP protocol
- Send Modbus commands to read or modify data on ICS devices
- Log output to console and/or JSON file

# MODBUS TCP PROTOCOL



# FrostyGoop

- Many binaries named modbus.exe
- Written in GoLang, use opensource libraries
  - [github.com/rolfl/modbus/](https://github.com/rolfl/modbus/)
  - [github.com/goccy/go-json](https://github.com/goccy/go-json)



```
rolfl
├── modbus
│   └── serial
│       ├── github_com_rolfl_modbus_ptr_client_query .text
│       ├── github_com_rolfl_modbus_ptr_client_query_func1 .text
│       ├── github_com_rolfl_modbus_X01xReadCoils_String .text
│       ├── github_com_rolfl_modbus_X05xWriteSingleCoil_String .text
│       ├── github_com_rolfl_modbus_X0FxFWriteMultipleCoils_String .text
│       ├── github_com_rolfl_modbus_X02xReadDiscretes_String .text
│       ├── github_com_rolfl_modbus_X14xReadRecordRequest_String .text
│       ├── github_com_rolfl_modbus_X14xReadFileRecordResult_String .text
│       ├── github_com_rolfl_modbus_X14xReadMultiFileRecord_String .text
│       ├── github_com_rolfl_modbus_X15xWriteFileRecordRequest_Str... .text
│       ├── github_com_rolfl_modbus_X15xWriteFileRecordResult_String .text
│       ├── github_com_rolfl_modbus_X15xMultiWriteFileRecord_String .text
│       └── github_com_rolfl_modbus_X03xReadHolding_String .text
```

# FROSTYGOOP FUNCTIONALITY

Information required to initiate a TCP connection and send Modbus commands to a victim ICS device can be specified in two ways:

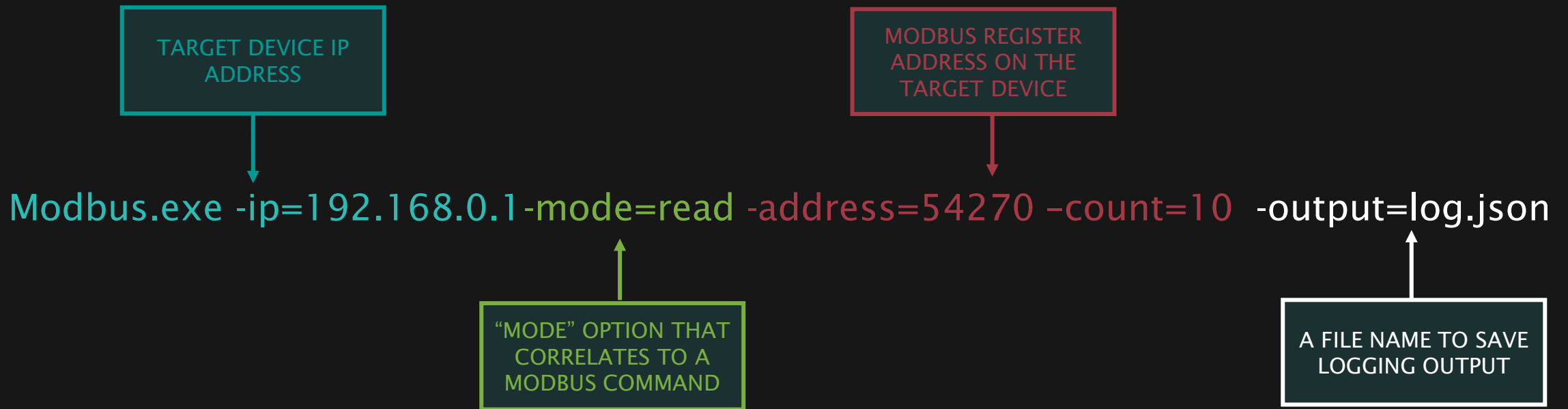
Command Line  
Arguments



JSON  
Configuration File



# FROSTYGOOP COMMAND LINE ARGUMENTS



|           |   |
|-----------|---|
| 'read'    | Command Code 3 'Read Holding Registers' - Read the value currently in a Modbus holding register (or contiguous block) |
| 'write'   | Command Code 6 'write Single Holding Register' - write a value to a holding register                                  |
| 'write-m' | Command Code 16 'write Multiple Holding Registers' - write a value to a block of contiguous registers                 |

# FROSTYGOOP JSON CONFIG

Modbus.exe -input-task=task.json

CONFIGURATION  
FILE NAME

```
{
  "iplist": ["192.168.0.1"],
  "Tasks": [
    {
      "Code": 3,
      "Address": 54270,
      "Count": 5,
      "Value": 0
    },
    {
      "Code": 3,
      "Address": 54275,
      "Count": 5,
      "Value": 0
    },
    {
      "Code": 6,
      "Address": 54280,
      "Count": 1,
      "Value": 1
    },
    {
      "Code": 16,
      "Address": 54285,
      "Count": 5,
      "Value": 1
    }
  ]
}
```

Command Code 3 'Read Holding Registers'

Read the value currently in a Modbus holding register (or contiguous block)

Command Code 6 'Write Single Holding Register'

Write a value to a holding register

Command Code 16 'Write Multiple Holding Registers'

Write a value to a block of contiguous registers

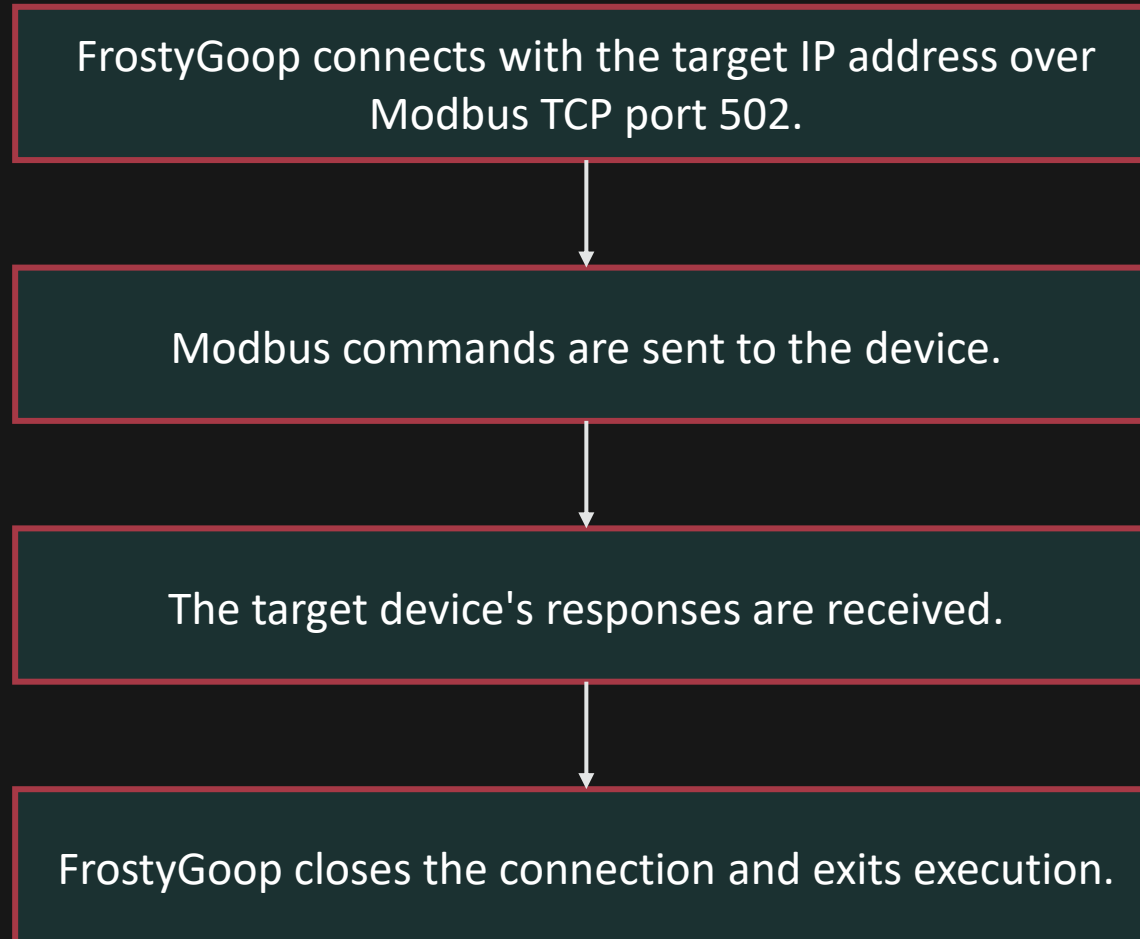
# FROSTYGOOP CONFIGURATION FILE

FrostyGoop accepts two different configuration files

Contains victim device information:  
IP address, Modbus commands, Modbus register addresses

Specifies start time for Modbus TCP communications and delay time for executing Modbus commands.

# MODBUS NETWORK TRAFFIC



# LOGGING CAPABILITIES

The FrostyGoop binaries log output from the Modbus TCP communications with the target IP address to a Windows console and a JSON file.

The FrostyGoop executable opens a Windows console upon execution. Below is an example of output logged to the console window during Modbus TCP communications with the target device.

If the argument for logging is specified when executing the binary, then the output is logged to a JSON file. Below is an example of the JSON log file.

```
[runtime.goexit:asn_and64.s:1598][INFO] | (1/1)
| start
| (1/1) | address: 53370 count: 5 + | 0s
| (1/1) | address: 53760 count: 10 + | 15.625ms
| (1/1) | address: 53882 value: 0 + | 0s
| (1/1) | address: 54272 count: 10 + | 15.625ms
[runtime.main:proc.go:250][INFO] Time delta | 2m3.5390625s
```

```
[{"Ip": "██████████", "Responses": [{"Fcode": 3, "Err": "", "Delta": 2234375000, "Time": "2024-07-10T14:24:04.3003984-04:00", "Response": {"Address": 53860, "Values": [1,1,1,1,1,1,1,1,1,1]}]}]}
```

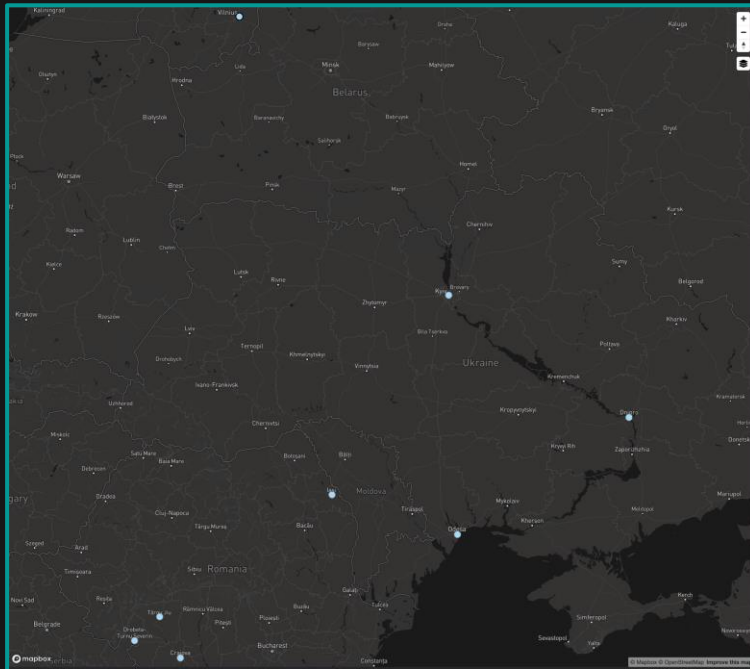
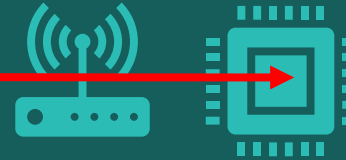
# TASK\_TEST.JSON

- Dragos discovered a sample of the configuration file named ‘task-test.json.’
- The IP address in the sample configuration file belongs to an ENCO control device.



- ENCO control devices are typically used “for process control in district heating, hot water, and ventilation systems” to monitor sensor parameters such as temperature, pressure, and insulation.

# Theoretical Attack



```
% nc -v 10.2.20.105 23
Connection to 10.2.20.105 port 23 [tcp/telnet] succeeded!
*====*
*          Enco control Telnet Server v1.00          *
*====*

Available Commands:
-----
ethr          - ethernet connection list
gprs          - gprs connection list
tcpconn      - tcp connections
ipstat       - IP statistics
icmpstat     - ICMP statistics
tcpstat      - TCP statistics
udpstat      - UDP statistics
owire        - one wire temperature sensors list
io           - inputs/outputs state
outX=Y       - change output X=(0 or 1) state Y=(0 or 1)
cport[=password,XXXX] - config port
rst=password - restart device
ntp          - ntp correction
help,?      - display this help
exit,<Ctrl+C> - disconnect

>
```

# ATTACK TIMELINE

**17<sup>th</sup> April 2023**  
Suspected  
compromise of  
Mikrotik Router  
vulnerability

**20<sup>th</sup> – 26<sup>th</sup> April 2023**  
Adversaries deploy  
and interact with  
webshell

**30<sup>th</sup> October 2023**  
FrostyGoop uploaded  
to public malware  
repository

**30<sup>th</sup> November 2023**  
Adversaries dumped  
SAM to gain  
credentials

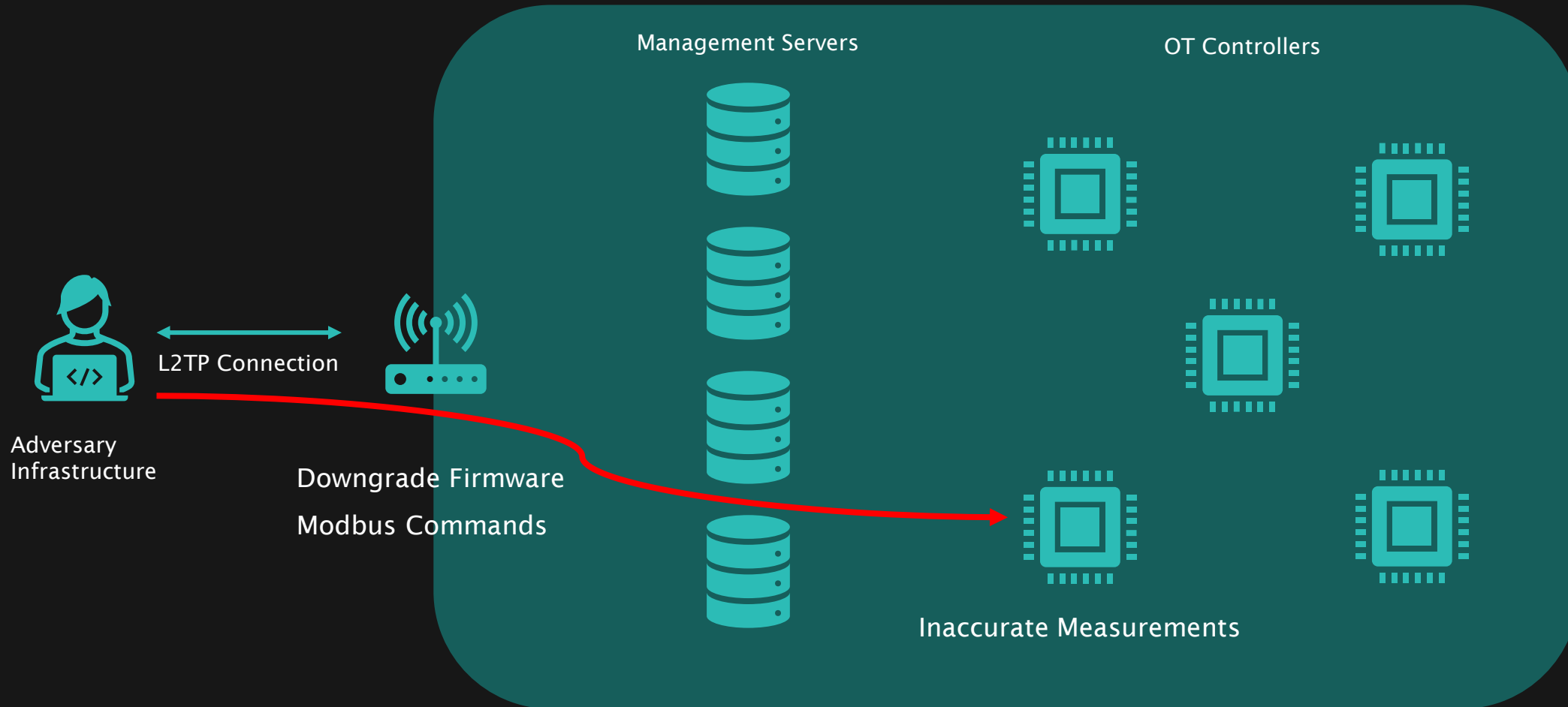
**14<sup>th</sup> December 2023**  
Adversaries dumped  
SAM to gain  
credentials again

**22<sup>nd</sup> January 2024**  
Adversaries initiate  
L2TP connection and  
conduct attack

Details shared by The Cyber Security Situation Center (CSSC), a part of the Security Service of Ukraine (SBU)



# The Attack Timeline

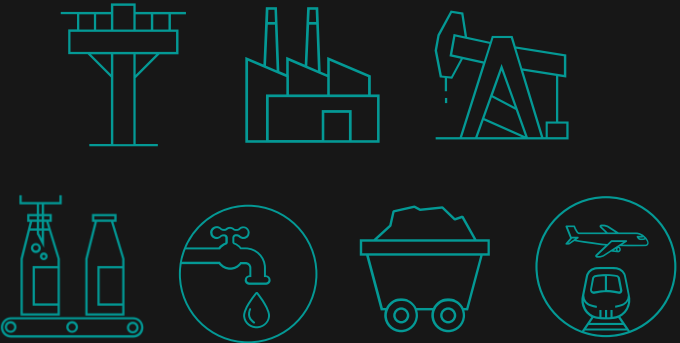


Details shared by The Cyber Security Situation Center (CSSC), a part of the Security Service of Ukraine (SBU)

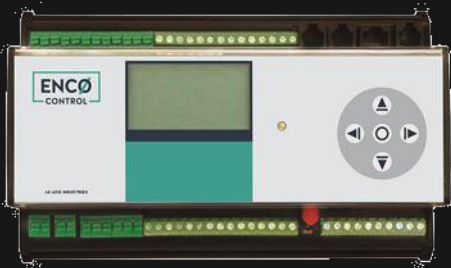
# CONNECTED ICS DEVICE EXPOSURE

**~46,000**

Internet-exposed ICS devices communicating over Modbus



*Modbus is used worldwide across industries.*



**~100**

ENCO devices exposed

# VULNERABLE PERIMETER DEVICES

- Exploiting vulnerable perimeter devices is not isolated to this event
- Threat groups and cybercriminals target edge devices for initial access

Threat Groups  
Targeting OT

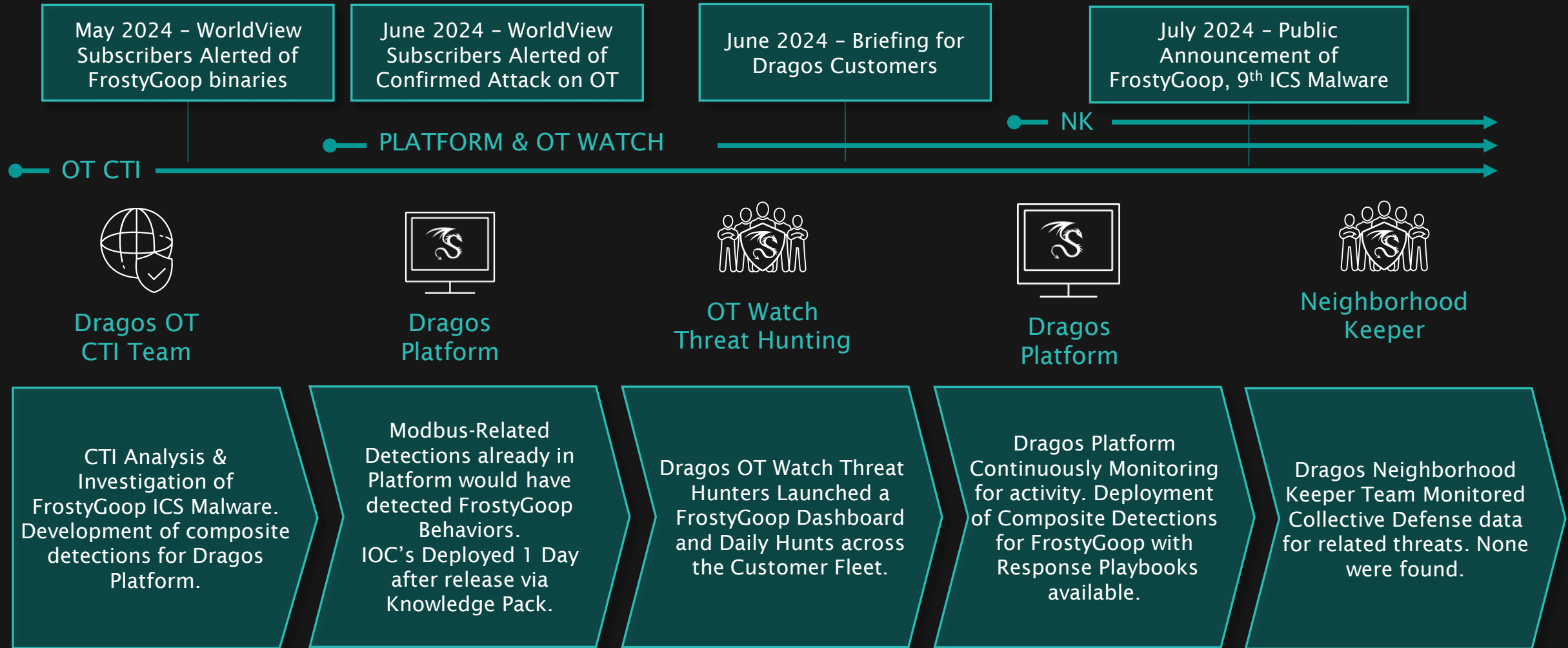
Hacktivist,  
Proxy-Hacktivist  
Groups

Ransomware,  
Cybercriminals

# RECOMMENDATIONS

- Utilize network segmentation, least privileges, and user access control practices to prevent adversary lateral movement in network environments.
- Ensure effective visibility and monitoring of North-South traffic in ICS locations to identify assets, connection details, and compromise attempts and allow faster survey and response during a cyber incident.
- Implement access controls to critical ICS/OT systems and devices, including restricting Modbus TCP port 502 access.
- Monitor devices for new connections on port 502.
- Conduct network telemetry analysis for unusual interactions with devices over port 502.
- Immediately identify and ensure ICS/OT systems and devices are inaccessible from the public-facing Internet.
- Keep perimeter devices updated with the latest security patches.
- If devices need to be accessible, place them behind firewalls or in a DMZ.
- Implement multi-factor authentication for users, particularly privileged users, and for remote access.

# POWER OF THE DRAGOS ECOSYSTEM



# TRANSFORM THREAT INTEL INTO DETECTIONS

**UNDERSTAND THE THREAT**  
BEHAVIOR, CAPABILITIES, INFRASTRUCTURE, INTENT

**OPERATIONAL CONSTRAINTS**  
PIVOT AGAINST BEHAVIORS & OPERATE WITHIN PLATFORM CAPABILITIES

| INITIAL ACCESS                      | EXECUTION                 | PERSISTENCE            | PRIVILEGE ESCALATION                  | EVASION                   | DISCOVERY                           | LATERAL MOVEMENT                 | COLLECTION                         | COMMAND & CONTROL                   | INHIBIT RESPONSE FUNCTION     | IMPAIR PROCESS CONTROL       | IMPACT                         |
|-------------------------------------|---------------------------|------------------------|---------------------------------------|---------------------------|-------------------------------------|----------------------------------|------------------------------------|-------------------------------------|-------------------------------|------------------------------|--------------------------------|
| Data Historian Compromise           | Change Operating System   | Modify Program         | Exploitation for Privilege Escalation | Change Operating Mode     | Network Connection Enumeration      | Default Credentials              | Automated Collection               | Commonly Used Port                  | Activate Firmware Update Mode | Brute Force I/O              | Damage to Property             |
| Drive-by Compromise                 | Command Line Interface    | Module Firmware        | Hooking                               | Exploitation for Evasion  | Network Sniffing                    | Establishment of Remote Services | Data from Information Repositories | Collection Proxy                    | Alarm Suppression             | Modify Parameter             | Denial of Control              |
| Engineering Workstation Compromise  | Execution Through API     | Project File Injection |                                       | Indicator Removal on Host | Remote System Discovery             | Lateral Tool Transfer            | Detect Operating System            | Standard Application Layer Protocol | Block Command Message         | Module Firmware              | Denial of View                 |
| Exploit Public Facing Application   | Graphical User Interface  | System Firmware        |                                       | Manipulating              | Remote System Information Discovery | Program Download                 | I/O Image                          |                                     | Block Reporting Message       | Spoof Reporting Message      | Loss of Availability           |
| Exploitation of Remote Services     | Hooking                   | Valid Accounts         |                                       | Rootkit                   | Wireless Sniffing                   | Remote Services                  | Man in the Middle                  |                                     | Block Serial COM              | Unauthorized Command Message | Loss of Control                |
| Internet Accessible Device          | Modify Controller Tasking |                        |                                       | Spoof Reporting Message   |                                     | Valid Accounts                   | Monitor Process State              |                                     | Data Observation              |                              | Loss of Productivity & Revenue |
| Remote Services                     | Native API                |                        |                                       |                           |                                     |                                  | Point & Tag Identification         |                                     | Denial of Service             |                              | Loss of Protection             |
| Replication Through Removable Media | Scripting                 |                        |                                       |                           |                                     |                                  | Program Upload                     |                                     | Detect Restart/Shutdown       |                              | Loss of Safety                 |
| Rogue Master                        | User Execution            |                        |                                       |                           |                                     |                                  | Screen Capture                     |                                     | Manipulate I/O Image          |                              | Loss of View                   |
| Spearfishing Attachment             |                           |                        |                                       |                           |                                     |                                  | Wireless Sniffing                  |                                     | Modify Alarm Settings         |                              | Manipulation of Camera         |
| Supply Chain Compromise             |                           |                        |                                       |                           |                                     |                                  |                                    |                                     | Rootkit                       |                              | Manipulation of View           |
| Wireless Compromise                 |                           |                        |                                       |                           |                                     |                                  |                                    |                                     | Service Stop                  |                              | Theft of Operational System    |
|                                     |                           |                        |                                       |                           |                                     |                                  |                                    |                                     | System Firmware               |                              |                                |

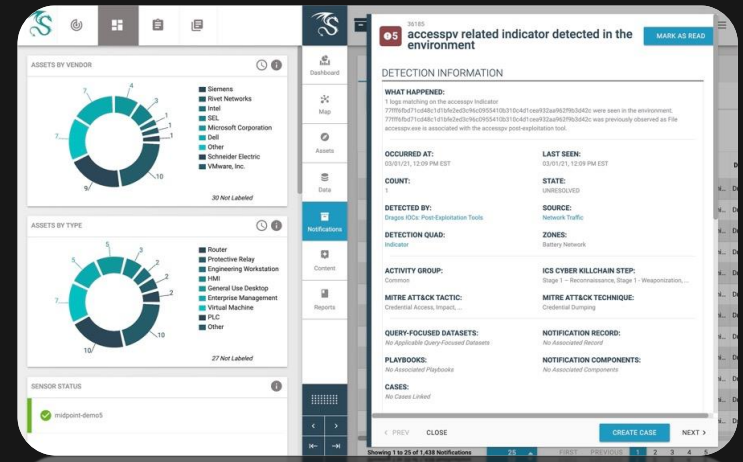
**Type**  
Indicator Configuration Modeling Threat Behavior

**Complexity**  
Atomic Composite

**Telemetry**  
Network Monitoring Host Logs

DATA SOURCES: DRAGOS THREAT INTELLIGENCE, OSINT RESEARCH, THIRD-PARTY THREAT INTELLIGENCE

DETECTIONS ARE CODIFIED IN THE DRAGOS PLATFORM KNOWLEDGE PACKS RELEASED REGULARLY WITH NEW THREAT INTELLIGENCE-DRIVEN DETECTIONS



# DRAGOS PLATFORM – EXISTING MODBUS DETECTIONS

| SID/Rule                             | Analytic Name                                | Description   | Knowledge Pack |
|--------------------------------------|--|---|----------------|
| a0ddb920-0adc-4d01-9b3d-21414ef28607 | Modbus Command Force Listen Only Mode        | Modbus command to put device into Force Listen Only Mode, making the device unresponsive to commands. It will only respond after power up. This can be used maliciously to effectively disable devices. | KP_Plus-7.0.X  |
| f7a0af6b-fa88-4382-9232-f56525befcde | Modbus Command Restart Communications Option | Modbus command to force a device to restart, making it unresponsive until it reboots. There is some chance this could be used maliciously to disable devices.   | KP_Plus-7.0.X  |
| f41c99e6-cabf-46b7-9576-d2ac4676baa9 | Modbus Exception                             | Modbus servers send exception codes to Modbus clients when a requested operation cannot be carried out. This characterization summarizes exception codes sent from a Modbus server.                     | KP_Plus-6.0.X  |
| e8cbde89-aa3a-4093-8064-3a8ca08fbf4c | Modbus External Comms                        | External device communicating with an internal asset using the Modbus protocol. This is a major security concern, as ICS devices should not be controlled outside of the OT network.                    | KP-2020-11     |
| 15c07ad4-5d03-4c3b-8d2d-613d5ec45217 | Modbus External Write                        | External device writing to an internal asset using the Modbus protocol. This is a major security concern, as ICS devices should not be controlled outside of the OT network.                            | KP-2020-11     |
| 3cc434cd-5086-454c-bbd4-6142b01a4623 | Modbus Write Observed for First Time         | Modbus traffic with a write function code seen for the first time to a specific host.   | KP-2022-009    |
| d323014b-abee-461b-a12f-641b8796070f | New ModbusTCP Detection                      | Monitors for new devices using the ModbusTCP protocol and generates events when activity is seen  | KP-2020-11     |

# FROSTYGOOP COMPOSITE DETECTION

**4** <sup>46</sup> FrostyGoop Malware Network Behaviors
ACTIONS ▾

---

### DETECTION INFORMATION

**WHAT HAPPENED:**  
Asset 7 using IP address 192.168.0.50 sent at least two (2) uniquely-formed Modbus TCP commands to asset 10 at IP address 192.168.0.7 within a time window of 60 seconds. The Modbus TCP commands sent by asset 7 were atypical because the network traffic resembled unique telemetry only produced by the FrostyGoop malware. Consult the Dragos Platform's playbook for FrostyGoop, linked in this notification, for ways to triage and respond to this alert. [◀ Read Less](#)

**OCCURRED AT:** 07/30/24, 04:57 PM CDT

**COUNT:** 1

**DETECTED BY:** FrostyGoop Behavior

**DETECTION QUAD:** Indicator

**THREAT GROUP:** N/A

**MITRE ATT&CK FOR ICS TACTIC:** Command And Control [🔗](#)

**QUERY-FOCUSED DATASETS:** No Applicable Query-Focused Datasets

**PLAYBOOKS:** No Associated Playbooks

**CASES:** No Cases Linked

**LAST SEEN:** 07/30/24, 04:57 PM CDT

**STATE:** UNRESOLVED

**SOURCE:** [ed965ff4-1ce2-4fe7-aa09-e89255bf9437](#)

**ZONES:** RFC1918

**ICS CYBER KILLCHAIN STEP:** Stage 2 - Install/Modify

**MITRE ATT&CK FOR ICS TECHNIQUE:** None

**NOTIFICATION RECORD:** [View in Kibana](#)

**NOTIFICATION COMPONENTS:** [View in Kibana](#)

### ASSOCIATED ASSETS

| View                 | Type        | ID | Criticality | Name     | Dir.             |
|----------------------|-------------|----|-------------|----------|------------------|
| <a href="#">VIEW</a> | General Use | 7  | —           | Asset 7  | 192.168.0.50 src |
| <a href="#">VIEW</a> | Asset       | 10 | —           | Asset 10 | 192.168.0.7 dst  |

### COMMUNICATIONS SUMMARY

Asset 192.168.0.50 ↔ MODBUS\_TCP / MODBUS ↔ Asset 192.168.0.7

| Proto... | Client       | Ephemeral Po...   | Server      | Server Ports | TX Bytes | RX Bytes |
|----------|--------------|-------------------|-------------|--------------|----------|----------|
| MODBU... | 192.168.0.50 | 49327,49328,49... | 192.168.0.7 | 502          | 1.4 KB   | 1.3 KB   |
| MODBUS   | 192.168.0.50 | 49327,49328,49... | 192.168.0.7 | 502          | 1.4 KB   | 1.3 KB   |

< PREV
NEXT >



# FROSTYGOOP PLAYBOOK – DRAGOS PLATFORM

The screenshot shows the Dragos Platform interface. At the top, there is a navigation bar with the Dragos logo and menu items: Detections, Health & Status, Cases, and Playbooks. The Playbooks menu is active. On the right side of the navigation bar, there are icons for a home page, a refresh icon, and a user profile labeled 'Administrator'. Below the navigation bar, the main content area displays the title 'FrostyGoop Malware Behaviors' with a star icon and a 'Dragos' tag. A 'BACK TO PLAYBOOKS' link is visible in the top left of the content area. On the right side of the content area, there are three action buttons: 'ADD TO CASE', 'EDIT', and 'EXPORT'. The main text area contains a paragraph stating that network traffic similar to that used by the FrostyGoop malware has been detected. Below this, there is a detailed explanation of why the alert fired, mentioning specific TCP patterns and ModbusTCP commands. A second paragraph explains the redundancy of the ModbusTCP unit identifier and provides context on a recent cyberattack in Ukraine. At the bottom of the text area, there is a list of links and report IDs. On the far right, there is a vertical sidebar with navigation icons for Dashboard, Map, Assets, Data, Notifications, Vulnerabilities, Reports, and Admin. At the bottom of the sidebar, there are navigation arrows and a task bar.

DRAGOS

Detections Health & Status Cases Playbooks Administrator

< BACK TO PLAYBOOKS

☆ FrostyGoop Malware Behaviors

Dragos

ADD TO CASE

EDIT

EXPORT

Network traffic similar to that used by the FrostyGoop malware has been detected in your environment.

This alert has fired because specific TCP patterns followed by legitimate ModbusTCP commands matching FrostyGoop behaviors have been detected in network traffic. FrostyGoop, the ninth ever ICS malware, is a sophisticated malware designed to interact directly with industrial control systems (ICS) using the ModbusTCP protocol. The Modbus and ModbusTCP protocols are both used for ICS communication and are some of the oldest and simplest ICS protocols in existence. The Modbus protocol and ModbusTCP differ primarily in their transport mechanisms and addressing methods. Modbus is a serial communication protocol that operates over RS-232, RS-485, or RS-422, using unit identifiers to address specific devices on the network. These unit identifiers are essential for routing messages to the correct device in a point-to-point or multi-drop setup. In contrast, ModbusTCP operates over Ethernet networks, encapsulating Modbus messages within TCP/IP packets. The addressing in ModbusTCP relies on IP addresses rather than unit identifiers.

The ModbusTCP unit identifier is largely redundant in ModbusTCP due to the use of IP-based communication (except when translating ModbusTCP to serial Modbus via a gateway). As written, FrostyGoop makes use of a Unit Identifier when sending ModbusTCP commands. It is able to manipulate system parameters, causing disruptions in critical infrastructure. FrostyGoop executes read and write commands against controllers, bypassing security measures by attempting to mimick legitimate traffic. FrostyGoop was used in a cyberattack against Ukraine in January 2024, specifically targeting Enco controllers to send unauthorized ModbusTCP commands that caused inaccurate measurements and system malfunctions. The attack resulted in a two-day disruption of heating services for a district energy company, impacting over 600 apartment buildings. For additional details on FrostyGoop's usage as well as its code functionality, see the following link and Dragos reports:

- (<https://www.dragos.com/resources/reports/intelligence-brief-impact-of-frostygoop-modbus-malware-on-connected-ot-systems/>)
- AA-2024-23
- AA-2024-19

TASKS

Dashboard

Map

Assets

Data

Notifications

Vulnerabilities

Reports

Admin

# RECOMMENDATIONS

SANS

5

THE FIVE  
ICS CYBER  
SECURITY  
CRITICAL  
CONTROLS

01

ICS Incident Response Plan

---

02

Defensible Architecture

---

03

ICS Network Visibility & Monitoring

---

04

Secure Remote Access

---

05

Risk-based Vulnerability Management

# Q&A

QUESTIONS AND ANSWERS