Trout

FrostyGoop Malware Report: A Comparative Analysis

A new approach to Industrial Networks Security

Executive Summary

This report provides a comparative analysis of the FrostyGoop malware incident, examining the findings from the Dragos report and the SCADASEC response. Dragos characterizes FrostyGoop as a sophisticated ICS-targeted attack by a Russian-linked group, whereas SCADASEC presents a differing perspective on the malware's attribution and sophistication, suggesting that further evidence may be needed to fully substantiate these claims.

Our independent analysis indicates that FrostyGoop lacks the advanced characteristics typically seen in state-sponsored malware and may not have been involved in the alleged attack. While FrostyGoop may possess some disruptive capabilities, it does not match the sophistication of other ICS-targeted malware, suggesting that its overall threat level should be revaluated. Key findings highlight areas where the evidence presented by Dragos might benefit from further clarification or additional context.

Trout

Introduction

Industrial Control Systems (ICS) are crucial for managing and automating essential processes in sectors such as energy, manufacturing, and utilities. As these environments become increasingly digitized, the risk of ICS-targeted malware continues to grow, posing significant threats to operational technology (OT). FrostyGoop, a malware allegedly designed to target ICS environments, has recently come under scrutiny following a controversial report by Dragos, which was countered by SCADASEC.

This report aims to critically evaluate the claims made by both Dragos and SCADASEC regarding FrostyGoop's capabilities and its potential involvement in a cyberattack on Lviv's heating infrastructure. By providing a detailed technical analysis and comparing the narratives presented by both sides, this report explores the broader implications for ICS cybersecurity and the necessity of further investigation into the true nature of FrostyGoop.

What is FrostyGoop?

FrostyGoop is a rare type of malware specifically designed to target ICS. According to Dragos, it is the ninth such malware, joining Trisis (Triton), CrashOverride (Industroyer), BlackEnergy2, Havex, Stuxnet, Industroyer2, PipeDream, and Fuxnet. Allegedly developed by the Sandworm team, a Russian state-sponsored APT group, FrostyGoop is believed to have been designed to disrupt OT by exploiting vulnerabilities in ICS networks.

Dragos claims FrostyGoop uses Modbus TCP communications, is written in Golang, and is compiled for Windows systems. This makes it particularly suited for ICS environments where these communication protocols and platforms are prevalent.

Incident Overview: The Attack on Lviv Heating Systems

In January 2024, Dragos reported a cyberattack on Lviv's heating infrastructure, allegedly carried out using the FrostyGoop malware. Dragos claimed that "Russian-linked malware was used in a January 2024 cyberattack to cut off the heating of over 600 apartment buildings in Lviv, Ukraine, for two days during sub-zero temperatures." The attackers reportedly gained initial access by exploiting vulnerabilities in externally facing routers and "downgraded the firmware on the ENCO controllers, deploying a version that lacks monitoring capabilities," leading to a heating disruption.

However, SCADASEC presents a different account based on official Ukrainian sources, stating that only 324 Individual Heating Units (IHUs) were affected, not 600 apartment buildings, and that "the heat supply was restored in 6 hours to 50% and 13 hours to 100%, not the 48 hours claimed in the report." SCADASEC also questions the role of ENCO devices in this incident. According to SCADASEC, ENCO devices are primarily used to read data from heat meters and deliver status data to a central server, with no evidence of their involvement in controlling physical processes. This function is supported by the Technical Specification Document, which outlines their purpose and capabilities.

Additionally, SCADASEC argues there is no mention of these devices supporting the Modbus protocol for control functions, although data from the meters could potentially be converted from Mbus to Modbus for transmission purposes—but only for transferring readings. This suggests, in SCADASEC's view, that ENCO devices, if involved at all, had a limited role that may not align with the attack scenario proposed by Dragos. Additional details about the types of ENCO devices sold in Ukraine, such as data loggers with GPRS modems, support this interpretation. For example, the Elmisto ENCO Device Listing shows these devices are primarily intended for data logging rather than direct control of heating systems.

SCADASEC also claims that there is no direct evidence confirming that ENCO devices were targeted or affected in the incident. They point out that the only link to ENCO devices comes from a hard-coded IP address in a configuration file found on VirusTotal, which points to an ENCO device in Romania—not Lviv. Furthermore, according to SCADASEC, there are no exposed ENCO devices in Lviv, based on Shodan scans, suggesting they were never online or exposed in this region SCADASEC Follow-Up Report.

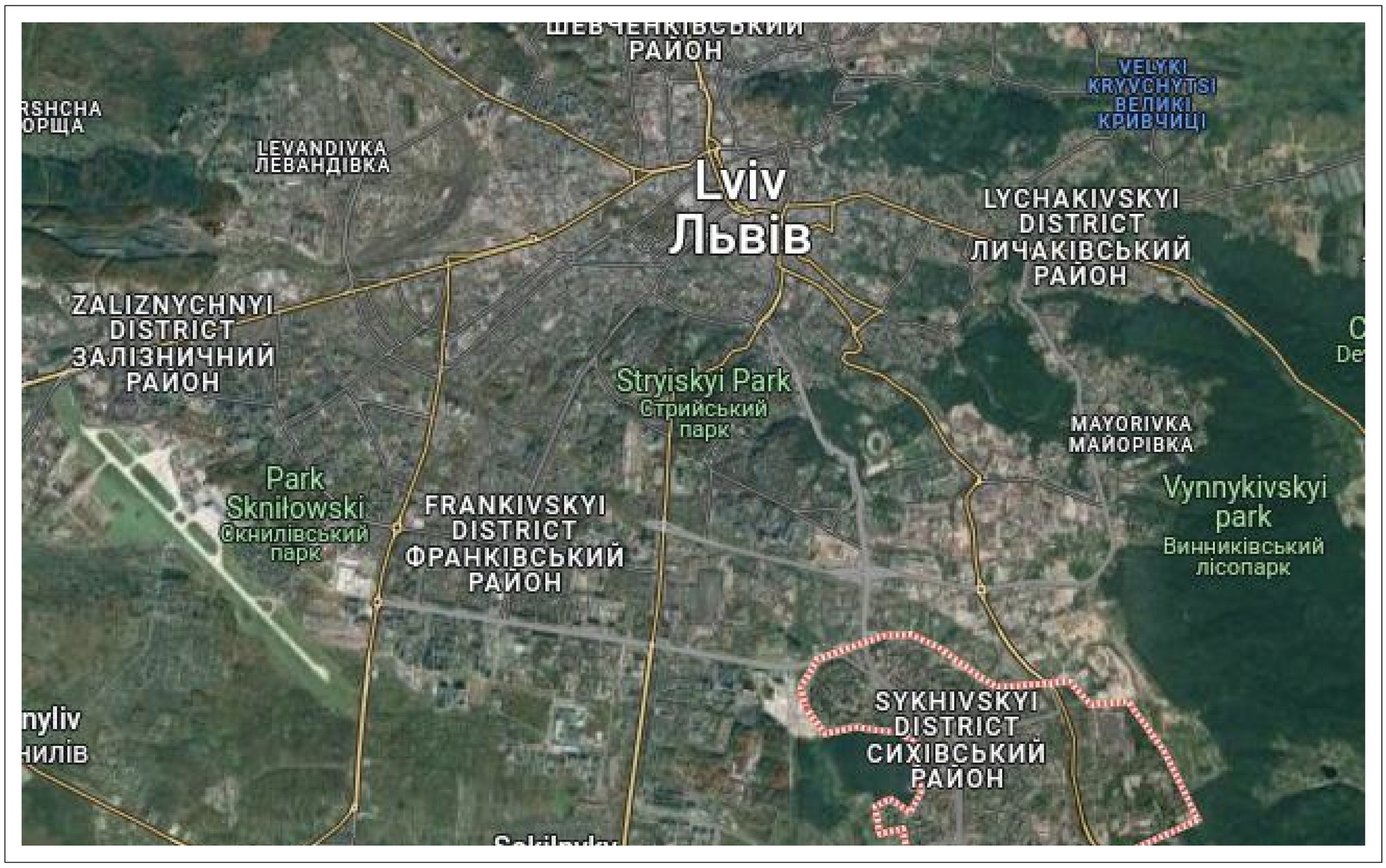


Figure 1: District of Sykhiv - Area allegedly affected by the heating disruption.

Figure 2: Lvivteploenergo - Company managing the heating infrastructure.

Lviv Mayor Andriy Sadovyi acknowledged the incident, describing it as a "malfunction," while adding, "there is a suspicion of external interference in the company's work system, which is currently being investigated." This ambiguity highlights the contested nature of the event and the differing narratives presented by Dragos and SCADASEC.

Technical Analysis of FrostyGoop

Summary of the Dragos Report

The Dragos report describes FrostyGoop's capabilities and its alleged role in the Lviv heating attack:

- Modbus TCP Communications: FrostyGoop utilizes Modbus TCP over the default port (502), a protocol commonly used in ICS environments. This represents a unique use case for malware within the ICS domain.
- Programming and Compilation: Written in Golang and compiled for Windows systems, FrostyGoop can operate in many ICS environments.
- Core Functionalities: The malware can read and write to ICS device registers, accept optional command-line execution arguments, and use configuration files to specify target IP addresses and Modbus commands. It logs its output to a console and/or a JSON file, potentially monitoring its impact on ICS devices.

Independent Analysis Methodology

To verify the claims made in the Dragos report, we conducted an independent analysis using the samples provided by VXunderground. Here it is :

Encrypted Files 1.zip

Encrypted Files 2.zip

Let’s analyze those samples and use Ghidra, the open source reverse engineering tool developed by the NSA. Our methodology included:

1. Sample Collection and Verification: The samples were downloaded from VXunderground, an educational repository for malware analysis. File hashes were verified against VirusTotal to ensure authenticity.
2. Static and Dynamic Analysis: Using tools like Ghidra, we performed static analysis to disassemble the binaries and identify key functions and behaviors. Dynamic analysis involved monitoring the malware’s execution in a controlled environment.
3. Reverse Engineering: In-depth reverse engineering with Ghidra focused on identifying the malware’s capabilities, potential targets, and any unique characteristics or patterns.

Sample Collection and Verification

When checked on VirusTotal, the file hashes do appear similar to the FrostyGoop malware:

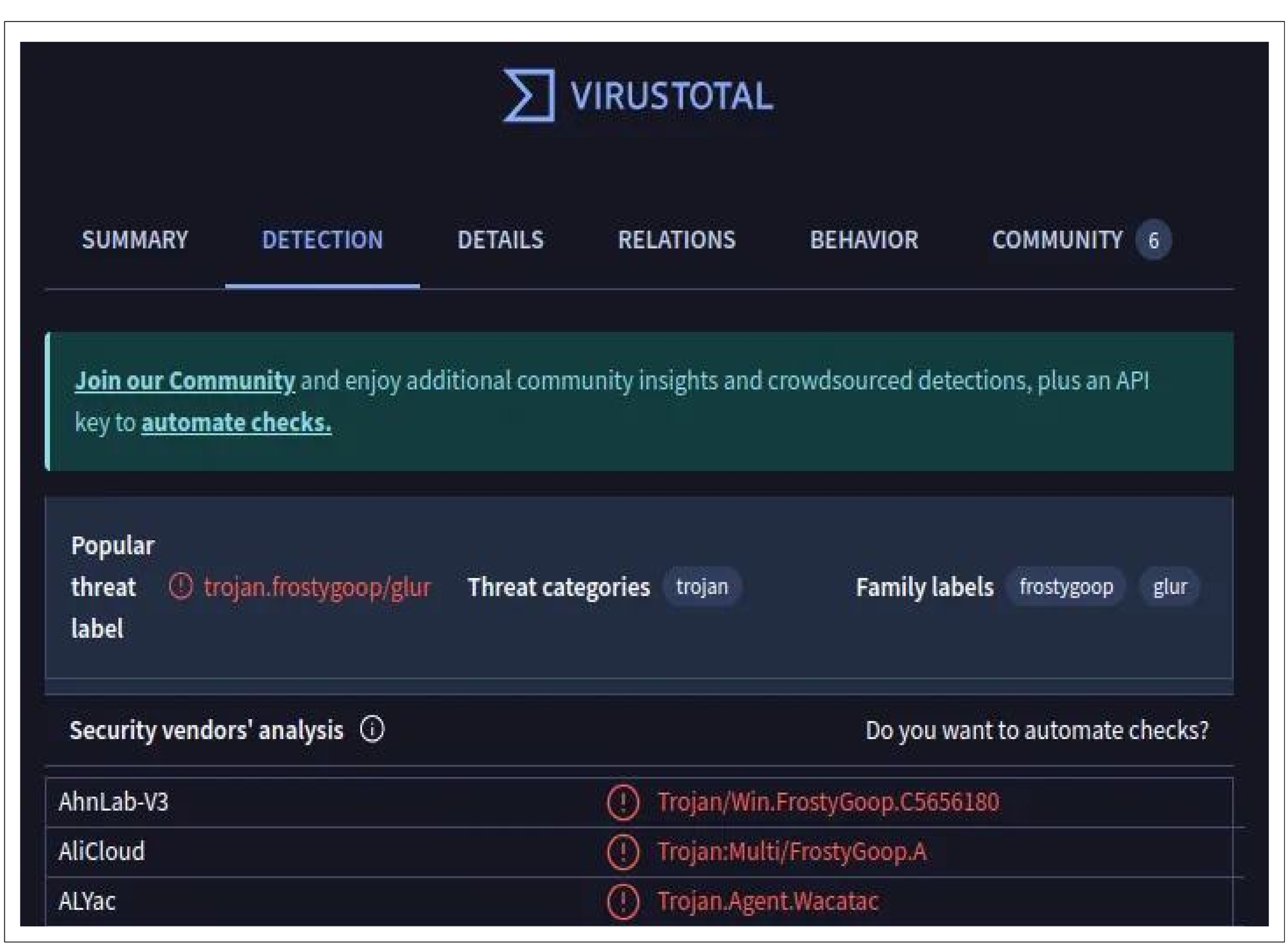


Figure 3: VirusTotal analysis showing detection results for FrostyGoop malware.

These files are indeed Windows executables:

```
CTI:~/Documents/FrostyGoop$ file 5d2e4fd08f81e3b2eb2f3eaae16eb32ae02e760afc36fa1
6da53fb.exe
fd08f81e3b2eb2f3eaae16eb32ae02e760afc36fa17f4649322f6da53fb.exe: PE32+ executable
) x86-64 (stripped to external PDB), for MS Windows, 13 sections
CTI:~/Documents/FrostyGoop$ file a63ba88ad869085f1625729708ba65e87f5b37d7be9153b
fed309c
88ad869085f1625729708ba65e87f5b37d7be9153b3db1a1b0e3fed309c: PE32+ executable (c
6-64 (stripped to external PDB), for MS Windows, 6 sections
```

Figure 4: Command-line output showing file details for Windows executables.

With a string or using the go version -m <executable> command, we can see that the Go language is used along with some Modbus capabilities due to the main module github.com/rolfl/modbus/ClientTCP. However, this alone is not enough to classify it as malware.

With a string or using the go version -m <executable> command, we can see that the Go language is used along with some Modbus capabilities due to the main module github.com/rolfl/modbus/ClientTCP. However, this alone is not enough to classify it as malware.

```
go version -m
5d2e4fd08f81e3b2eb2f3eaae16eb32ae02e760afc36fa17f4649322f6da53fb.exe
5d2e4fd08f81e3b2eb2f3eaae16eb32ae02e760afc36fa17f4649322f6da53fb.exe:
go1.20.4
 path github.com/rolfl/modbus/CleintTCP
 mod github.com/rolfl/modbus (devel)
 dep github.com/hsblhsn/queues v0.0.0-20220219165404-d2097de75d81
h1:E/7K5MuiqpTABTG9N9yFzH38Z+R6/o7KszaVVzOZXEc=
 dep gopkg.in/logex.v1 v1.1.10
h1:wspNZImtG1i5tkn3LLhr9nSls8+JZZgDfv6+pAs36hY=
 build -buildmode=exe
 build -compiler=gc
 build CGO_ENABLED=0
 build GOARCH=amd64
 build GOOS=windows
 build GOAMD64=v1
```

Static and Dynamic Analysis

To evaluate the claims made about FrostyGoop, we conducted a detailed analysis using Ghidra, an open-source reverse-engineering tool developed by the NSA. Ghidra provides a deep dive into the binary, allowing us to assess its functionality and sophistication.

Our analysis of the FrostyGoop samples yielded several observations that could be crucial in understanding the malware's functionality.:

- **Lack of Obfuscation:** The binary shows no signs of obfuscation, a technique commonly used by advanced malware to evade detection. This absence is unusual for malware attributed to a state-sponsored group like a Russian APT, suggesting either a lower level of sophistication or a different strategic intent by the attackers.

- **Basic Functionality as a Modbus Client:**

```
                    ****************************************************************
                    * DWARF original prototype: void main.main(void)               *
                    * Golang function info: Flags: [], ID: NORMAL                  *
                    * Golang source: C:/Users/Hiro Kirashi/Documents/Projects...   *
                    * Golang recovered signature: undefined main.main()            *
                    ****************************************************************
                    void abi-internal main.main(void)
                       assume R14 = 0xfffffffffffeffa98
                       assume XMM15 = 0x0
    void              <VOID>          <RETURN>
    undefined8        Stack[-0x8]:8  local_8                          XREF[3]:     00521d19(W),
                                                                                   00521d21(*),
                                                                                   00522e4f(*)
    undefined1[16]    Stack[-0x18]... local_18                        XREF[2,1]:   00522505(W),
                                                                                   0052253d(R),
                                                                                   0052250d(W)
    undefined1[16]    Stack[-0x28]... local_28                        XREF[2,1]:   005224f5(W),
                                                                                   00522531(R),
                                                                                   005224fd(W)
    undefined1[16]    Stack[-0x38]... local_38                        XREF[3,1]:   005224e5(W),
                                                                                   0052251e(R),
                                                                                   0052255a(*),
                                                                                   005224ed(W)
    undefined1[16]    Stack[-0x48]... local_48                        XREF[2,1]:   00522885(W),
                                                                                   005228bd(R),
```

Figure 5: Disassembled code in Ghidra showing basic Modbus client functionality.

The malware appears to function primarily as a generic Modbus client. It is capable of reading and writing analog outputs (e.g., 0-100% values) but lacks the ability to interact with digital inputs/outputs or manipulate specific ICS processes. This limited functionality indicates that the malware may not have the advanced capabilities typically seen in ICS-targeted attacks, such as those targeting safety instrumented systems (SIS) or causing physical damage.

- **No Indicators of Malicious Libraries:** The executable does not depend on libraries typically used in advanced malware, such as those for network attacks, encrypted communication, or exploitation frameworks. Instead, the build settings are standard, with no signs of tampering or evasion techniques. This further supports the notion that FrostyGoop lacks the sophistication expected of state-sponsored malware.

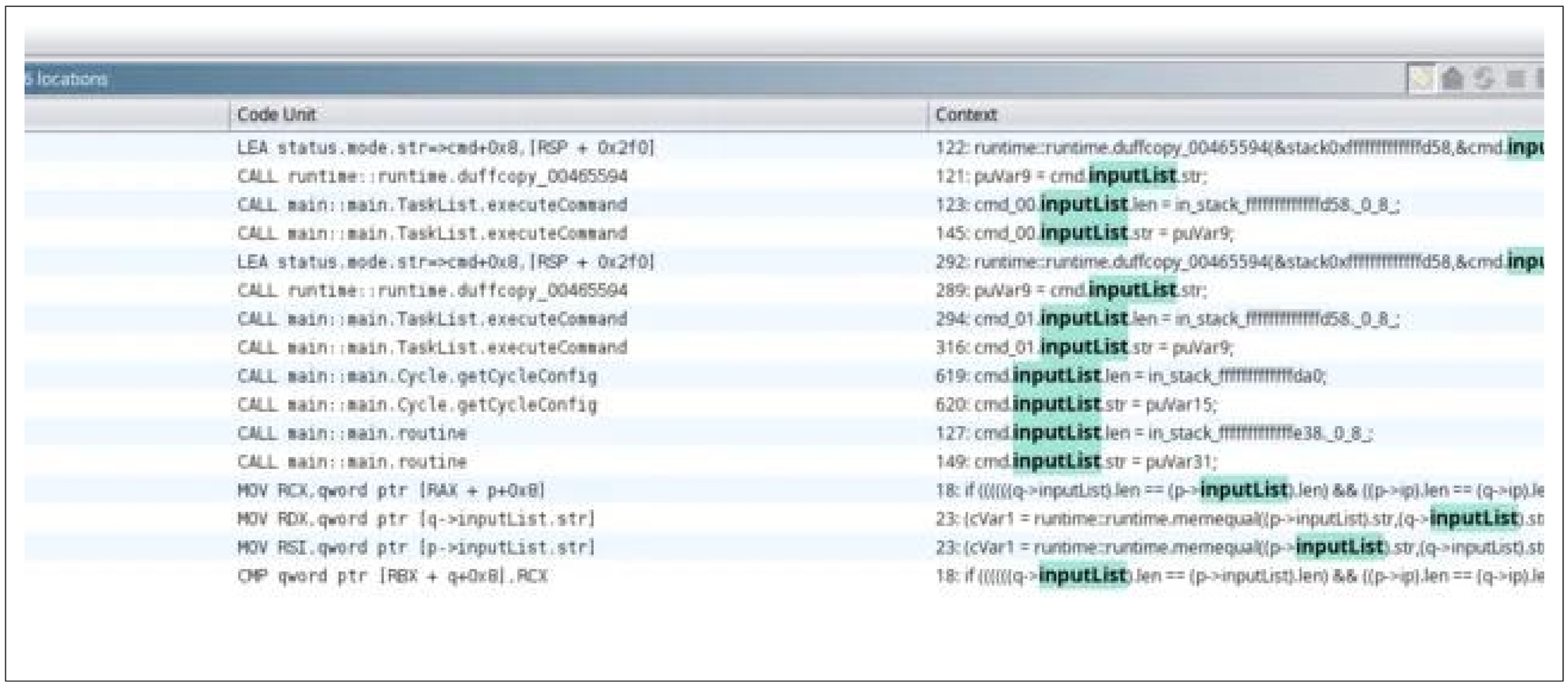


Figure 6: Detailed Ghidra analysis output indicating the lack of malicious libraries.

The most obvious fact that catches our eye is that there is zero obfuscation. We can see who compiled the binary, and it is clearly unprofessional for Russian APT-related activity.

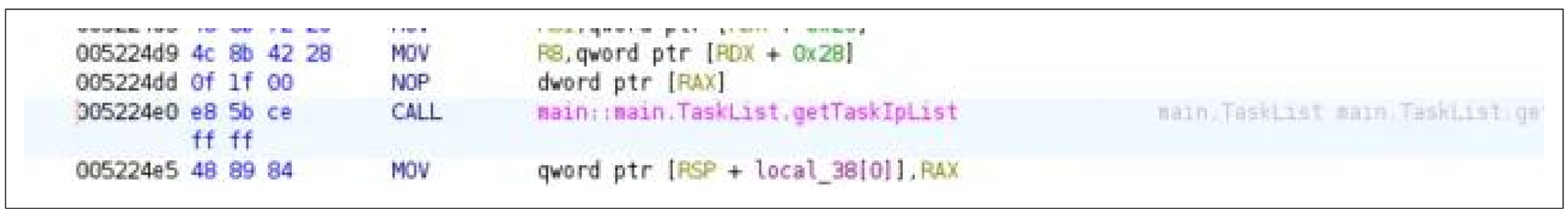


Figure 7: Further analysis showing that the binary is a simple Modbus client.

The analysis indicates that the sample appears more akin to a generic Modbus client, which may limit its capability as a fully developed malware tool.

The binary lacking the advanced features and obfuscation techniques typically associated with state-sponsored APT malware, our analysis did not find sufficient evidence to support the specific claims made in the Dragos report.

Additionally, the malware appears to be more of a generic tool than a specialized attack framework. Marina Krotofil, a respected expert in ICS security, supports this view, stating:

> *"The discovered sample is a generic Modbus client capable of reading and writing analog outputs (basically 0-100% values). Full stop. It is not programmed to interact with digital inputs/outputs... It is a big stretch to say that the tester of the malware sample was interested in ENCO devices and that the sample had something to do with the incident in Ukraine."*

Krotofil's insights align with our findings, suggesting that the connection between the malware and the alleged attack on Lviv's heating systems is tenuous at best. The lack of obfuscation, limited functionality, and absence of malicious libraries point to a tool that is more likely a low-level experiment than a state-sponsored weapon. Instead, it appears to be a generic Modbus client with limited capabilities. This calls into question the narrative presented in the Dragos report and highlights the need for further investigation to determine the true nature and origin of FrostyGoop.

So, What Can We Trust?

Although the possibility of a malware attack cannot be completely dismissed, the available evidence does not substantiate claims about the involvement of ENCO devices or the scale of the reported attack. No direct proof of malware has been found, nor is there any indication that an attack on ENCO devices could have caused the severe consequences described in Dragos's report.

However, this incident is a good reminder of the need for robust security hygiene in ICS environments, such as securing internet-exposed devices, implementing network segmentation, and applying the mitigations outlined below. With conflicting reports and a lack of corroborating evidence, it's essential to critically assess the information at hand and avoid drawing conclusions without adequate proof.

Dr. Marina Krotofil also emphasizes the importance of avoiding unsupported assumptions:

> *"It's not our role to make insinuations about the situation without clear evidence. We should focus on the facts at hand and ensure that any conclusions are based on solid data."*

MITRE ATT&CK Mapping and Cyber Kill Chain Analysis

The MITRE ATT&CK framework and the Cyber Kill Chain provide structured ways to analyze potential tactics, techniques, and procedures (TTPs) that could be associated with the FrostyGoop malware, as described in the Dragos report. The analysis here is speculative, based on the possible attack vectors and outcomes indicated in the reports, and aims to provide a comprehensive understanding of how such an attack could unfold.

Tactics and Techniques

Tactic	Technique ID	Technique Name	Context
Reconnaissance (Initial Access)	T0883	Internet Accessible Device	Exploiting ICS devices directly accessible over the internet due to open ports or weak network security.
Reconnaissance (Initial Access)	T0866	Exploitation of Remote Services	Exploiting vulnerabilities in remotely accessible services (e.g., routers, VPNs, firewalls).
Weaponization	-	No specific techniques identified	Lacks sophisticated weaponization characteristics (e.g., custom payloads, zero-day exploits).
Delivery	T0812 / T0859	Default Credentials / Valid Accounts	Using default credentials or valid accounts to move laterally within the ICS environment.
Exploitation	T0821	Modify Controller Tasking	Modifying ICS device registers to manipulate the tasking of controllers (e.g., PLCs).

Tactic	Technique ID	Technique Name	Context
Exploitation	T0855	Unauthorized Command Message	Sending unauthorized Modbus commands to ICS devices to alter their functionality.
Installation	T0835	Manipulate I/O Image	Manipulating I/O images to alter the perceived state of the ICS system.
Post-Exploitation (Command and Control)	T0815	Denial of View	Downgrading firmware or corrupting data to blind operators to the actual state of the ICS environment.
Post-Exploitation (Command and Control)	T0813	Denial of Control	Taking over control commands to prevent operators from managing ICS devices.
Actions on Objectives	T0826	Loss of Availability	Disrupting critical services (e.g., heating) by shutting down or degrading ICS systems.
Actions on Objectives	T0880	Loss of Safety	Manipulating ICS devices to cause unsafe operating conditions, potentially leading to harm.

Breakdown by Cyber Kill Chain Phases

- **Reconnaissance (Initial Access):**
  - **Internet Accessible Device (T0883):** Attackers likely used open ports or misconfigured devices, such as ENCO controllers, to gain initial access.
  - **Exploitation of Remote Services (T0866)**: Attackers could have exploited vulnerabilities in routers, VPNs, or other remote services to infiltrate the ICS network.

- **Weaponization:**
    - **No specific techniques identified:** The FrostyGoop malware lacks the characteristics of advanced weaponization seen in more sophisticated attacks, such as Stuxnet or Triton.
- **Delivery:**
    - **Default Credentials (T0812) / Valid Accounts (T0859):** Attackers might use weak or stolen credentials to move laterally, positioning themselves for further exploitation.
- **Exploitation:**
    - **Modify Controller Tasking (T0821):** By reading and writing to ICS device registers, the malware could alter normal operations, potentially causing disruptions.
    - **Unauthorized Command Message (T0855):** Sending rogue Modbus commands to manipulate device behavior.
- **Installation:**
    - **Manipulate I/O Image (T0835):** Altering input/output images to mislead operators or automated systems, concealing the attacker's presence.
- **Post-Exploitation (Command and Control):**
    - **Denial of View (T0815):** Obstructing monitoring by downgrading firmware or corrupting data streams.
    - **Denial of Control (T0813):** Seizing control of commands to prevent legitimate operator interventions.
- **Actions on Objectives:**
    - **Loss of Availability (T0826):** Aiming to disrupt services (e.g., heating) by degrading the performance of ICS systems.
    - **Loss of Safety (T0880):** Manipulating safety mechanisms to create hazardous situations, impacting both human safety and physical assets.

Mitigations

To defend against attacks like the hypothesized FrostyGoop attack, organizations should consider the following mitigations:

- **Network Segmentation:**
  - Implement network segmentation to isolate critical ICS networks from other networks (IT, other iOT, public wifi...). This limits an attacker's ability to move laterally, reducing the impact of a potential compromise.

- **Network Monitoring:**
  - Continuously monitor network traffic for unusual patterns, such as unauthorized Modbus commands or unexpected data flows. Use intrusion detection and prevention systems (IDPS) to detect and alert on suspicious activities.

- **Demilitarized LAN (DLAN):**

  DLAN improves upon traditional network segmentation and facilitate micro segmentation by deploying small, software-defined DMZs in front of each LAN. This approach offers several advantages:

  1. Simplified Implementation: DLAN scales effectively to protect large numbers of devices by creating isolated, secure zones for each machine using a software-defined, zero-trust architecture.
  2. Enhanced Security: DLAN combines firewall, proxy, and NAT functions to provide layered protection and visibility into network traffic, while also acting as a certificate authority to ensure comprehensive encryption and secure communications.
  3. Scalable Monitoring and Compliance: DLAN enables conitunous monitoring of activities, spanning across layers 2 to 7, and serves as a compliance checkpoint to enforce policies, and supports the deployment of secure enclaves within a network.

- **Strong Access Control and MFA:**
  - Enforce strict access control measures, including multi-factor authentication (MFA), to protect sensitive ICS devices and networks. Regularly update passwords and avoid using default or weak credentials.

- **Incident Response:**
    - Develop and maintain an Incident Response Plan (IRP) tailored to ICS environments. Train all relevant stakeholders to ensure quick and effective responses to incidents.

- **SOC, SIEM, and EDR Solutions:**

  Deploy a comprehensive suite of security tools:

    - SOC (Security Operations Center): Provides continuous monitoring, threat detection, and rapid response capabilities.
    - SIEM (Security Information and Event Management): Aggregates and analyzes log data from various sources, providing insights into potential threats.
    - EDR (Endpoint Detection and Response): Monitors and responds to threats at the endpoint level, including ICS devices.

Comparative Analysis with Known ICS Malware

While FrostyGoop is classified as ICS-targeted malware, it lacks many of the advanced features that characterize some of the most significant malware threats in this space. Comparing FrostyGoop with well-known ICS malware like Stuxnet, Triton, and CrashOverride helps to highlight the gaps in its sophistication and potential impact.

Significant Characteristics of Advanced ICS Malware

Advanced ICS malware typically exhibits several key characteristics that make it highly effective and dangerous:

- **Multi-Stage Payloads:** These involve deploying malware in multiple stages, with each stage designed to achieve specific objectives (e.g., initial reconnaissance, lateral movement, payload delivery, and command-and-control). Multi-stage payloads increase the stealth and flexibility of the malware, making detection and mitigation more difficult.
- **Self-Propagating Mechanisms:** Some ICS malware can autonomously spread across networks, exploiting vulnerabilities in connected devices or systems. Self-propagation enables rapid infection of multiple systems, leading to widespread disruption.

- **Zero-Day Exploits:** Advanced ICS malware often employs zero-day exploits — vulnerabilities that are unknown to the software vendor and for which no patch is available. This allows the malware to bypass conventional security measures and directly target specific systems or equipment.

- **Protocol Manipulation and Specificity:** Effective ICS-targeted malware is often designed to manipulate specific industrial protocols or devices, such as programmable logic controllers (PLCs), in ways that cause physical damage or operational disruptions.

Comparison of FrostyGoop with Known ICS Malware

- **Stuxnet:**

  Stuxnet is widely recognized as one of the most sophisticated ICS-targeted malware ever developed. It utilized multiple zero-day exploits to infect specific Siemens PLCs (Programmable Logic Controllers) used in Iran's nuclear enrichment facilities. Stuxnet employed a multi-stage payload, enabling it to stealthily gain access, spread across networks, reprogram controllers, and sabotage centrifuge operations by manipulating their rotational speeds. These features allowed Stuxnet to evade detection for a significant period and cause physical damage to critical industrial equipment.

  - Why FrostyGoop Falls Short: FrostyGoop lacks several of the critical features that made Stuxnet so effective. Unlike Stuxnet, which had self-propagating mechanisms to spread across networks autonomously, FrostyGoop relies on manual deployment, limiting its ability to infect multiple systems quickly. Moreover, FrostyGoop does not have multi-stage payloads or zero-day exploits, which are essential for conducting sophisticated and undetectable attacks. Instead, FrostyGoop's reliance on a single protocol (Modbus TCP) and its basic functionality as a Modbus client restrict its impact to simple read and write operations, making it incapable of complex, targeted attacks or causing physical damage to critical infrastructure.

- **Trisis (Triton):**

  Triton specifically targeted safety instrumented systems (SIS) at a petrochemical plant, aiming to cause physical damage by manipulating the safety controls designed to prevent hazardous conditions. Triton's payload was tailored to compromise Triconex SIS controllers, potentially leading to catastrophic failures and even loss of life. Triton's ability to interact with these highly specialized systems demonstrated a deep understanding of industrial safety processes and advanced capabilities for manipulating specific hardware and software configurations.

- Why FrostyGoop Falls Short: FrostyGoop does not have specialized payloads or the capability to target and manipulate SIS or other critical safety controls. Its basic Modbus client functionality lacks the depth needed to compromise specific hardware or control systems in a way that could lead to significant physical harm or operational disruptions.

- **CrashOverride (Industroyer):**

  CrashOverride, also known as Industroyer, was designed to target electrical substations by manipulating multiple industrial communication protocols, such as IEC 104 and IEC 61850. The malware had modules specifically crafted to communicate with and control different types of ICS devices. CrashOverride's ability to disrupt power distribution by manipulating these protocols demonstrated a deep knowledge of the grid's operational technology and a tailored approach to causing widespread outages.

  - Why FrostyGoop Falls Short: In contrast, FrostyGoop only utilizes the Modbus TCP protocol, which limits its applicability to a narrower range of ICS environments. It does not have the versatility or depth needed to affect multiple protocols or device types within an ICS network, reducing its potential impact and effectiveness.

Key Takeaways

The characteristics of FrostyGoop, such as its lack of multi-stage payloads, self-propagating mechanisms, and protocol manipulation specificity, may suggest a different threat profile compared to other well-known ICS-targeted malware like Stuxnet, Triton, and CrashOverride. While it may have some disruptive capabilities in an unprotected environment, its overall lack of sophistication suggests it is not in the same category as these more advanced threats.

Further investigation is necessary to determine whether FrostyGoop represents a novel, less sophisticated threat actor's attempt at targeting ICS, or if it is simply a low-level experimental tool with limited real-world impact.

SCADASEC's Counterpoints and Further Analysis

According to SCADASEC, "the discovered sample is a generic Modbus client capable of reading and writing analog outputs (basically 0-100% values)." This aligns with our findings, which suggest the malware is more of a basic tool than a sophisticated weapon. SCADASEC also highlights that the connection to ENCO devices is tenuous, noting, "the only relation to ENCO devices... is the IP address in Romania," which casts doubt on the malware's involvement in the Lviv incident.

Notes of caution

While further investigation is required to fully understand its origin and intent, this analysis highlights the need for critical scrutiny in cybersecurity reporting.

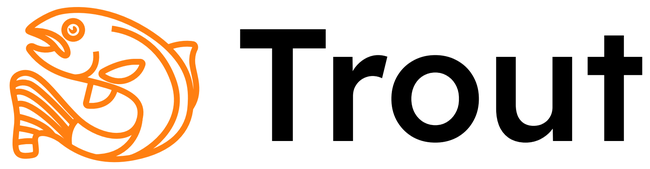
However, this incident serves as an important reminder of the need for strong security measures, including securing internet-facing devices, enforcing network segmentation, and implementing the mitigations discussed below.

Appendices

- **Technical Details**: Additional data from malware analysis, including sample hashes and VirusTotal results.
- **References:** Links to original reports, articles, and official statements.
- **Glossary:** Definitions of technical terms used in the report.

References

- **Dragos Report on FrostyGoop**
- **SCADASEC Response to Dragos**
- **Malware Sample Analysis**


Trout