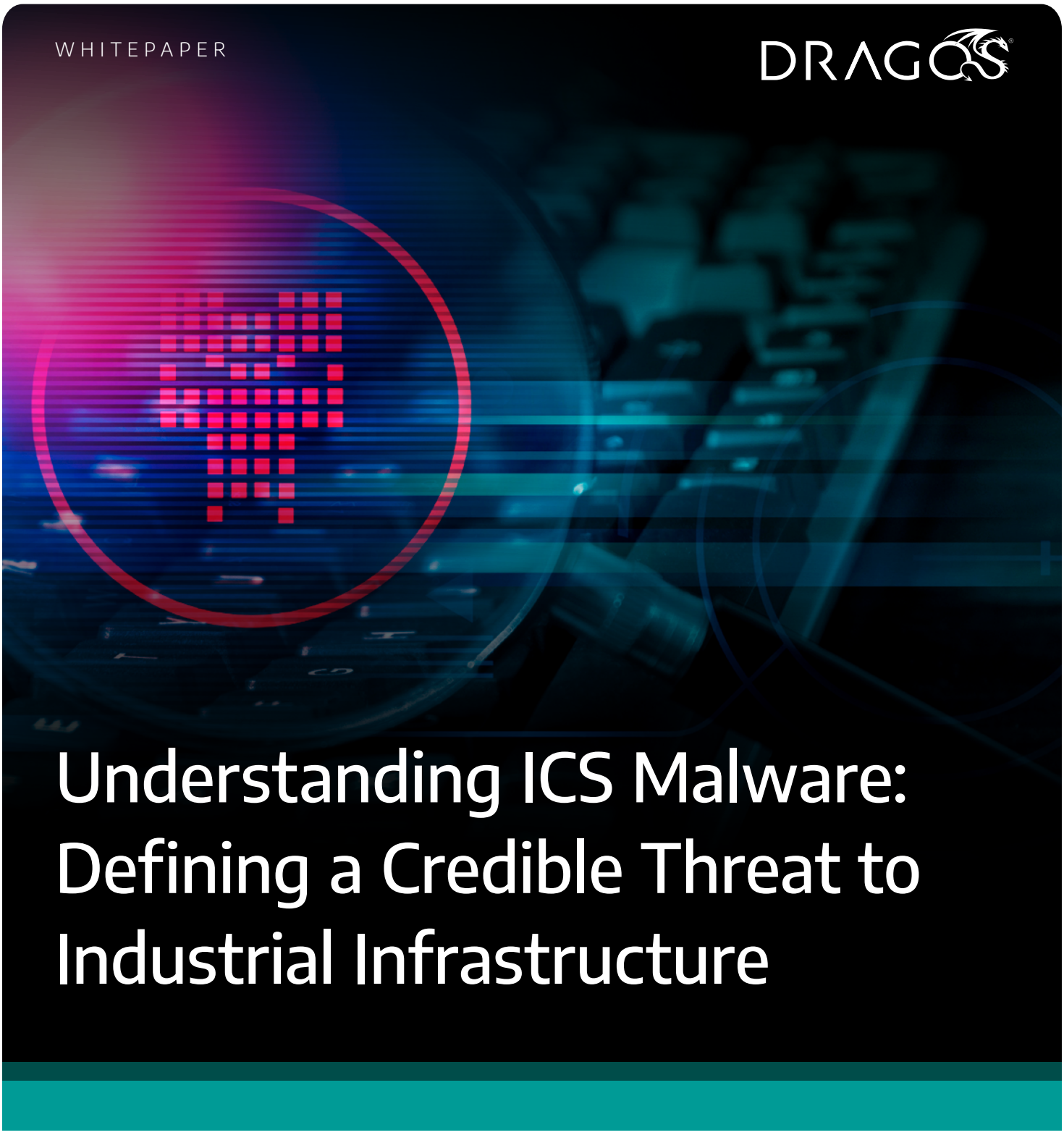
WHITEPAPER



Understanding ICS Malware: Defining a Credible Threat to Industrial Infrastructure

JIMMY WYLIE | DISTINGUISHED MALWARE ANALYST

DRAGOS, INC

JUNE 2025

TABLE OF CONTENTS

TABLE OF CONTENTS

A Definition of ICS Malware: ICS-Capable, Malicious Intent, and Adverse Effects

Introduction

The Industrial Control Systems (ICS) Malware Definition is a method for identifying and analyzing ICS malware. The definition requires that a candidate sample have three properties: ICS-Capability, Malicious Intent, and the Ability for Adverse Effects on operational technology (OT) environments[1]. This paper outlines the definition, the three properties, and how to differentiate ICS malware from other types of software and malware. Using TRISIS, IoT Exploit Tool, and IOControl as case studies, the paper shows how to apply the three properties to identify ICS malware as part of an intelligence-based process.

ICS Malware Definition

ICS malware is ICS-capable software intentionally designed for adverse effects on OT environments. To be considered ICS malware:

- ☑ The software must be ICS-capable.
- ☑ The software must be designed with malicious intent.
- ☑ The software must have the ability for adverse effects on OT environments.

Identification of ICS malware is an evidence-based intelligence assessment. If sufficient evidence for all three properties is found, the ICS Malware designator is applied to the sample.

Background and Motivation

In 2010, the community uncovered the first ICS malware sample, Stuxnet. At that time, the ICS community understood there was an innate difference between IT malware and ICS malware due to the potential for physical effects on an industrial process. In the years leading up to 2020, the ICS malware that followed —Havex, BlackEnergy2, CRASHOVERRIDE, and TRISIS—were, like Stuxnet, self-evidently ICS malware. A formal definition of ICS malware was unnecessary.

As the community's visibility and hunting efforts increased, new malware challenged that innate understanding of ICS malware. In 2020, EKANS Ransomware targeted industrial organizations and had the capability to terminate ICS-related Windows processes[2]. It was unclear at the time whether that ability made EKANS ICS malware. At customer sites, Dragos often encountered situations where a file infector virus would indiscriminately infect executables on a system including an engineering software update. Later, the update would be transferred to an OT host. While Dragos would not classify this case as ICS malware, it highlighted the need to consider previously overlooked factors, such as the adversary's intent, when making an assessment.

In 2024, Dragos discovered the FrostyGoop ICS malware, which used Modbus TCP as part of an attack that disrupted heat to hundreds of apartment buildings in Ukraine[3]. After releasing details on the malware, Dragos received questions about why FrostyGoop was considered ICS malware but not other offensive Modbus TCP research. These questions highlighted gaps in the community's understanding of how Dragos identifies ICS malware.

1 In this paper, the author uses ICS and OT interchangeably to describe the non-IT industrial networks and environments that control and monitor physical processes. The industry appears to be gravitating towards OT as the primary term, but there is no standard, and the distinction does not affect this work in a meaningful way.
2 Mysterious New Ransomware Targets Industrial Control Systems - Wired
3 Impact of Frostygoop Modbus Malware on Connected OT Systems

To solve this problem, Dragos created a definition of ICS malware based on historical data and the company's collective experience responding to and analyzing malware affecting OT. To guide Dragos's threat intelligence work, the definition captures key features of ICS malware while retaining simplicity for ease of understanding. This definition provides assurance that what Dragos calls ICS malware is, in fact, software that presents a credible, real-world threat to critical infrastructure.

Caveat: ICS Malware is a Small Subset of ICS Threats

Exploitation of vulnerabilities, living off-the-land techniques, ransomware, and other IT malware targeting OT are all threats to OT environments. There are countless examples of these types of threats but only nine examples of ICS malware. As of the publishing of this paper, those nine ICS malware samples are: Stuxnet, Havex, BlackEnergy2, CRASHOVERRIDE, TRISIS, Industroyer2, PIPEDREAM, Fuxnet, and FrostyGoop.

This paper focuses on distinguishing ICS malware from other threats to OT, but this does not imply these other ICS threats are any less serious or merit reduced attention. Dragos tracks all credible cyber threats to critical infrastructure, regardless of category. However, ICS malware can have different effects against OT environments than other OT threats, and it is important for scenario-based defensive planning to prepare for an ICS malware attack differently. To do so, it helps to have a consistent method for identifying ICS malware and a shared language to discuss it.

Three Properties of ICS Malware

ICS-Capable

ICS-capable, taken from the SANS Institute[4], means the software contains ICS/OT functions for navigating, altering, or retrieving information from OT networks, devices, or software. In other words, its code allows for speaking ICS protocols or interacting with PLCs and other OT devices. Here are a few examples of ICS-capable software:

- Software that speaks ICS protocols like IEC104, OPC UA, and HART.
- Software that can upload or download ladder logic programs.
- Software that runs on the PLC interacting with ladder logic runtime or other operating system internals.
- Software that runs on the engineering workstation that interacts with or modifies the engineering software (e.g., modifies project files).

The ICS-capable property distinguishes the subset of ICS software from other types of software like standard Windows or Linux tools. Tools like Process Hacker, day-to-day Windows and Linux malware implants, and current ransomware variants are not considered ICS malware because they are not ICS-capable.

Software can be ICS-capable irrespective of its reputation. Rockwell Automation's Studio 5000 is ICS-capable software for managing and configuring Rockwell Automation PLCs[5]. TRISIS is ICS-capable malware for exploiting Schneider Electric Triconex Safety Systems[6]. While ICS-capable software can be malicious or used for malicious purposes, it is not necessarily malicious.

4 SANS uses this term in their courses and material like 2022 SANS ICS Security Summit
5 Studio 5000 Logix Designer
6 TRISIS: Analyzing Safety System Targeting Malware - Dragos

Designed with Malicious Intent

Designed with Malicious Intent means the software was intentionally designed to cause harm or negative consequences to OT environments. The shorthand for this property is Malicious Intent.

Malicious intent is important for distinguishing malware from defender tools and other tools designed for a benign purpose. In the IT space, PSExec, a standard admin tool, can be used by adversaries for lateral movement. Still, PSExec is not malware because it was designed for system administrators[7]. Similarly, in the ICS space, something that looks like ICS malware but does not have malicious intent could be ICS research, an ICS red team tool, or a subcomponent of a vendor's engineering software abused for malicious purposes.

To determine if the software was designed with malicious intent, Dragos uses evidence like code capabilities and behavior, binary similarity, developer information, threat group association, incident response data, and victim/deployment information.

The Ability for Adverse Effects on OT Environments

The **Ability for Adverse Effects on OT Environments** means the software works correctly to achieve negative outcomes against the OT environment (hosts, network, ICS process, etc.). This property introduces a burden of proof on the analyst to show the software can achieve an adverse outcome and specifies the category of actions considered detrimental for OT. Identifying this property answers the question: What adverse consequences can the ICS-capable software cause?

Adverse effects are actions that enable espionage, access, manipulation, disruption, damage, or any other outcome that furthers adversary objectives in the OT environment. Adverse effects are like those described as Stage 2 effects in the ICS Cyber Kill Chain and vary by site and industry[8]. Here are a few examples of adverse effects:

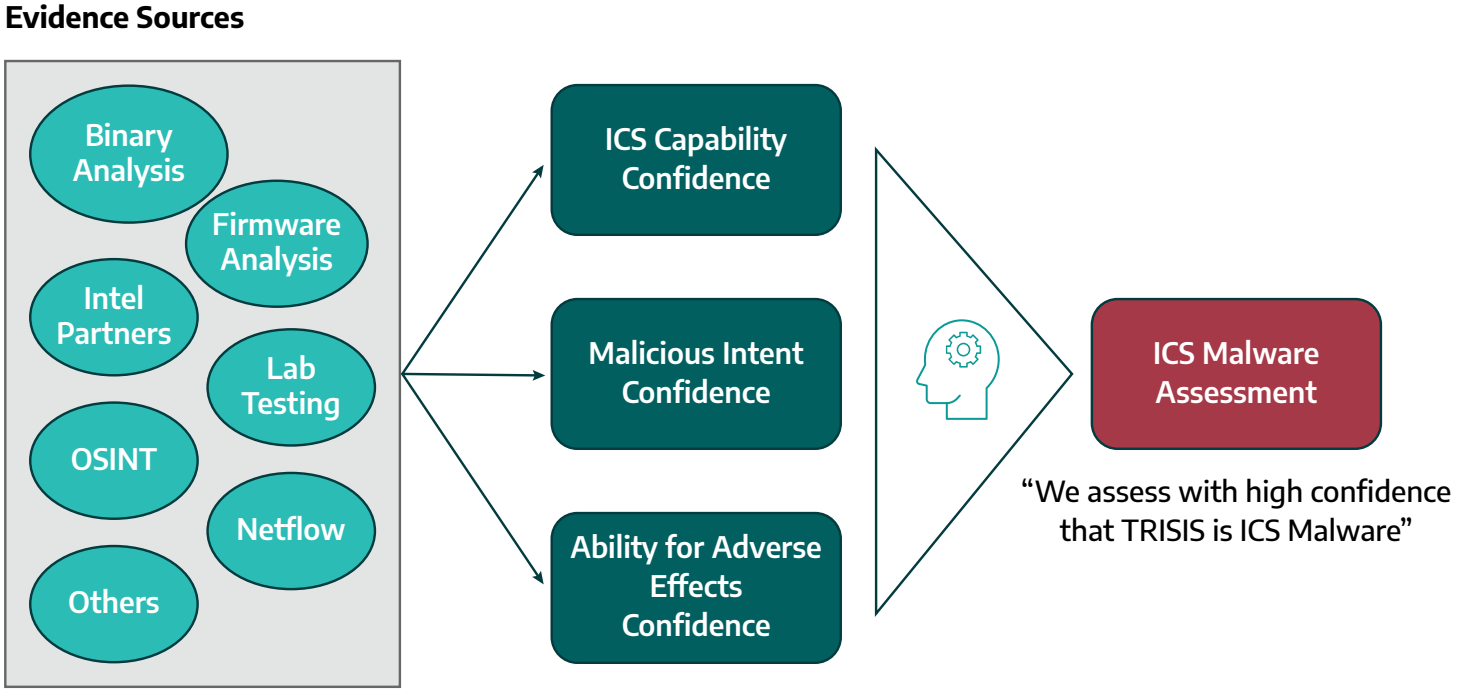
- Collecting sensitive process information.
- Enabling unauthorized access to OT devices.
- Downloading arbitrary ladder logic or executable code to PLCs.
- Bypassing OT firewalls or other OT security controls
- Manipulation of set points, variables, or other PLC settings.

What if the malware does not work? If a tool is ICS-capable and designed with malicious intent but has no ability for adverse effects, it is likely broken malware or malware in development. Depending on how close it is to functioning, Dragos may call it potential ICS malware.

7 PSExec - Microsoft Learn
8 The Industrial Control System Cyber Kill Chain - SANS

Identifying ICS Malware is an Intelligence Assessment

For ease of understanding, the definition is stated plainly as if it and its properties are self-evident when looking at a sample. However, in practice, when identifying ICS malware, analysts are faced with making an assessment based on imperfect information from multiple sources. This task is well suited to intelligence analysis, where pieces of evidence are weighed against the quality of their sources to produce guidance. Therefore, Dragos considers the identification of ICS malware as an evidence-based intelligence assessment.



ICS malware as an Intelligence Assessment

For each property of ICS malware, the analyst considers each piece of evidence along with its quality and source and determines with confidence whether a particular property holds for the tool: Dragos assesses with high confidence that TRISIS was maliciously designed for OT. All three assessments are combined to assess ICS malware: Dragos assesses with high confidence that TRISIS is ICS malware.

Speaking in confidence levels and assessments outside of an intelligence context is cumbersome, so in regular conversation, that assessment becomes TRISIS is ICS malware. In other words, any statement that identifies a sample as ICS malware is shorthand for an underlying confidence assessment.

Combining the Properties - Possible Outcomes

Software can have adverse effects and malicious intent without being ICS-capable, as is the case with much IT malware. For example, Windows ransomware could encrypt Windows hosts in an OT environment. This disruption has an adverse effect on the OT environment, but the malware is not ICS-capable, so the malware is not ICS malware. Instead, Dragos would refer to it as "Windows ransomware in an OT environment."[9] Software can be ICS-capable and have the ability for adverse effects but lack malicious intent, which is the case for red team tools. Software can be ICS-capable and have malicious intent but lack the ability for adverse effects, as is the case for broken ICS malware.

9 Pedantically, IT malware has general malicious intent but not always malicious intent for the OT environment, which is key for making an assessment. Depending on targeting, IT malware may lack ICS-capability and malicious intent for OT environments, disqualifying it doubly

These examples are a small subset of the possible combinations of properties. The following table lists all combinations and their likely outcomes.

ICS-Capable	Malicious Intent	Ability for Adverse Effects on OT	Result
✓	✓	✓	ICS malware
✓	✓	✗	Broken tool or evidence of ICS malware in development
✓	✗	✓	Offensive research, red team tool, or ICS/OT software with potential for abuse
✓	✗	✗	ICS/OT software or research
✗	✓	✓	ICS-themed malware or IT malware used in OT intrusions
✗	✓	✗	Broken or in development IT malware intended for OT intrusions[10]
✗	✗	✓	IT software potentially abused in OT
✗	✗	✗	Benign software

The Relationship Between ICS-Capable and Adverse Effects

When assessing ICS malware, there should be a relationship between the ICS-capable code and the ability for adverse effects. This relationship affects whether Dragos labels a sample ICS malware, and which parts receive the label in the case of multi-component malware. For demonstration, the examples in this section assume the malicious intent property.

Consider a typical file infector virus that has the following behavior:

- It embeds its code inside an executable.
- It modifies the executable so that, on launch, the malicious code will execute first.
- Upon execution, it opens a backdoor and then launches the original executable.

This virus infects ICS-capable engineering software. The newly formed, infected software is seemingly ICS-capable and has the ability for an adverse effect on OT, allowing unauthorized access via a backdoor.

The key question becomes: Is the virus code self-contained and independent of the victim software? Put another way, does the virus only use the victim software as a disguise to gain execution without using any of the victim software's ICS-capable code?

10 In cases where there is general malicious intent but not malicious intent for OT, this is probably an incidental or legacy malware infection.

Suppose the virus does not need to be inside an ICS-capable binary to function and could open a backdoor using any victim executable. In that case, there is no relationship between the ICS-capable code in the infected binary and the virus's ability for adverse effects. The virus does not use the available ICS-capable code, so the virus is not ICS malware based on a lack of ICS-capability.

However, the opposite conclusion is true if the virus uses underlying ICS-capable code inside the victim executable. For instance, the virus opens a backdoor and uses the engineering software's Modbus TCP library to send control commands given by the attacker. Now, there is an explicit relationship between the virus's adverse effects code and the ICS-capable code in the original executable. With this relationship established, the virus demonstrates the ICS-capable property. The virus is ICS malware.

Similar logic would guide the assessment of malware plugin frameworks. For example, an IT-focused malware framework like MATA features self-contained, independent plugins that only rely on the base malware for initial execution.[11] If MATA had a plugin for opening a backdoor to dispatch Modbus TCP commands, the plugin would contain the code for opening the backdoor, the Modbus TCP library, and the requisite logic for translating attacker commands into Modbus function codes. This means that the plugin would have both the ability for adverse effects and the ICS-capable code required for executing that ability, so Dragos would apply the ICS Malware designator to the plugin. However, Dragos would likely avoid using the designation for the whole MATA framework since the base malware and other plugins are not ICS-capable.

Conversely, if the malware framework exported ICS-capable functionality for the plugins to use, then there would be a link between the ICS-capable code in the base malware and the adverse effects caused by the plugins. In the EVILSCHOLAR malware from PIPEDREAM, the plugins implement attacks like a CODESYS Denial of Service using CODESYS protocol code inside the base malware.[12] The framework demonstrates the three properties, and there is a relationship between the ICS-capable code in the base and the adverse effects of the plugins. The entire framework is ICS Malware.[13]

Identifying ICS Malware: Three Case Studies

This section contains three real-world case studies for applying the definition in the identification of ICS malware:

- TRISIS (2017): ICS malware that attacked a Schneider Electric SIS[14]
- IoT Exploit Tool (2023): An ICS/OT and IoT red team tool for finding vulnerabilities[15]
- IOControl (2024): A Linux backdoor that infected Orpak fuel management devices and others[16]

TRISIS

In mid-November 2017, Dragos discovered TRISIS was deployed against one victim in the Middle East. Later investigation and reporting confirmed that the victim was an oil and gas facility in Saudi Arabia.[17] Dragos calls the threat group behind this attack XENOTIME.[18]

The intrusion resulted in multiple site shutdowns due to malfunctions in the safety instrumented system (SIS or safety controller). The incident response data confirmed that, throughout the intrusion, the attackers downloaded multiple versions of the TRISIS malware to workstations, and the malware was assessed as the probable cause of the shutdowns.[19]

11 MATA: Multi-platform targeted malware framework - Kaspersky
12 Analyzing PIPEDREAM: Results from Runtime Testing - Dragos
13 There is a potential third case: a malware framework where the base malware is not ICS-capable but where most of the plugins are independent ICS plugins. Each plugin would be considered ICS Malware. But, since the entire toolkit's purpose would be to affect ICS, for practicality and communication's sake, Dragos would refer to the whole framework as ICS Malware.
14 TRISIS: Analyzing Safety System Targeting Malware - Dragos
15 Page 48 in the Dragos 2025 OT Cybersecurity Report is the only public reference to this tool.
16 Page 26 - Dragos 2025 OT Cybersecurity Report
17 TRISIS has the security world spooked, stumped and searching for answers - Cyberscoop
18 XENOTIME - Dragos
19 TRITON - A Report From The Trenches - S4 2019

TRISIS targets the Schneider Electric Triconex Safety System with 3008 Main Processor boards. Using the proprietary TriStation 1131 Protocol, the malware downloads an exploit, a zero-day at the time, to the safety controller, and installs a rootkit. The rootkit functions as an additional network command. This network command allows for arbitrary read/write and execution of code and bypasses the key switch protection, which is intended to prevent code changes when set to RUN mode.[20]

TRISIS is an ideal example of ICS malware. Using details from the malware and the corresponding attack, this case study applies the ICS Malware definition by sorting the main facts into the three properties of ICS Capability, Malicious Intent, and Adverse Effects against OT environments.

ICS Malware Property	Evidence
ICS-Capable	Yes. Speaks TriStation protocol, can download code, PPC binaries with firmware addresses.
Malicious Intent	Yes. Only known deployment is part of an attack in 2017, Contains rootkit to bypass key switch protections.
Adverse Effects on OT	Yes. Caused multiple site shutdowns. Testing confirmed the network command allows access to the SIS regardless of key switch position and the ability to read/write/execute. Replicated SIS crash in the lab.
ICS Malware? Yes	

ICS-Capable: Yes

The TRISIS malware features four primary components: a Python-compiled executable named trilog.exe, an external Library.zip file with supporting Python libraries, and two PowerPC binary files. Both the Python and PowerPC components are ICS-capable.

The Python portions of TRISIS demonstrate ICS-capability by implementing and using the TriStation Protocol, a proprietary protocol only used by Triconex industrial safety systems and related software. Library.zip provides Python functions that use the TriStation protocol for downloading, uploading, and enumerating programs and functions on the safety controller. Using these functions, Trilog.exe composes a program from both PowerPC binaries and downloads it to the controller.

The PowerPC binaries of TRISIS demonstrate ICS-capability in their ability to navigate and modify the firmware operating system of the SIS. They contain hardcoded addresses from the firmware for details like the network command handler and system call handler that support the implementation of the zero-day exploit and rootkit command handler installer. Further, the logic of the command handler demonstrates an understanding of the underlying TriStation packet structure, the interface between command handlers and the firmware, and which regions of memory have executable permissions.

Malicious Intent: Yes

Victim and deployment information provide evidence for malicious intent. The 2017 intrusion against an oil and gas facility is the only known deployment of TRISIS, where it was used for unauthorized access and manipulation of an SIS. An attacker deployed TRISIS on an engineering workstation in a position to cause a specific effect, disruptive or otherwise, and TRISIS was assessed as the probable cause of the site's outages.

20 Analyzing TRISIS - the first Safety Instrumented System Malware - RECON 2018

There is no reasonable benign justification for TRISIS's code capability, which subverts the SIS's safety and security, and potentially endangers the associated ICS process. TRISIS uses a 0-day exploit to install a rootkit that allows an attacker to ignore the SIS's key switch protections and standard code download procedures to arbitrarily read, write, and execute unverified code. The Python pieces alone can achieve a plant shutdown with a direct control command, or a simple logic download. The rootkit is not necessary to achieve those effects, so its existence suggests an adversary intent on specific process impacts beyond a shutdown.

```
        case MP_EXECUTE:
            //make sure our packet is the right size
            //Headers + mpcmd, mpid, and address
            if (ts_len < 16)
                goto END;
            //make sure we are executing address
            //from executable region
            if (pPacket->mp_exec.addr >= 0x00100000)
                goto END;
            void (*foo)(void) = pPacket->mp_exec.addr;
            foo(); //execute address

            //only need to send back headers
            reply_len = 10;
            break;
        default:
            break;
    }

    //set reply command code
    pPacket->ts_cmd = MP_STATUS_REPLY;
    //set length in packet
    pPacket->ts_len = reply_len;
    //send reply packet (I think)
    send_ts_reply_68F0C();

    END:
        branchToNetworkCommandReturn();
}
```

Decompiled excerpt of TRISIS's rootkit network command for executing arbitrary code

Other intelligence information, such as the threat group — XENOTIME has a history of ICS-targeting operations starting in 2014 — and corroboration with trusted intel sources and partners, would be part of this assessment. However, the forensic evidence from the intrusion, the intrusion's effects, and the code capabilities of TRISIS are more than sufficient to conclude that TRISIS contains malicious intent for OT environments.

Ability for Adverse Effects on OT Environments: Yes

The ability for adverse effects against OT environments was demonstrated during the attack and in a lab. During the 2017 intrusion, multiple site shutdowns occurred, with TRISIS assessed as the cause. Dragos analysts successfully executed TRISIS and installed the network command rootkit on a Triconex Safety system in a lab. Further, Dragos analysts used the rootkit to read and write memory and crash the controller while the key switch was in the RUN position.

Aside from shutdown effects, the rootkit provides privileged access to the firmware. With this level of access, an attacker can overwrite any part of the system to hide malicious code, evade detection mechanisms, and manipulate running programs and outputs.[21] All these actions are considered adverse effects against OT environments.

Result

TRISIS has all three properties the definition requires, so it is ICS malware.[22] It speaks TriStation protocol, can download code to the SIS, and includes PowerPC binaries with embedded firmware knowledge, so TRISIS is ICS-capable. The only known deployment of TRISIS was in an attack against an oil and gas facility and it includes a rootkit for bypassing the Triconex SIS key switch protections, so TRISIS has malicious intent for OT environments. Since it caused multiple site shutdowns, and Dragos analysts were able to replicate that functionality in the lab and intentionally crash the SIS, TRISIS is proven to cause adverse effects against OT Environments.

21 TRISIS Takeaways: Defensive Techniques and Trench-Building for the Blue Team - Dragos
22 If this were an intelligence report, this sentence would read, "Dragos assesses with high confidence that TRISIS is ICS malware."

IoT Exploit Tool

In 2023, Dragos discovered the IoT Exploit Tool while routinely hunting for compiled Python binaries with ICS-related strings. Written in Python and transpiled to C via Cython, the tool contains over a thousand publicly available exploits targeting IP Cameras, NVRs, DVRs, routers, and industrial devices. It features many Chinese strings, one of which identifies it as an "IoT device vulnerability scanning, verification, and exploitation toolkit."

In the ICS category, IoT Exploit Tool has roughly 175 exploits for publicly disclosed vulnerabilities in products from common ICS/OT manufacturers such as Schneider Electric Modicon, Siemens SIMATIC S7, Emerson Ovation, and Rockwell Automation MicroLogix. It provides at least partial support for 19 industrial protocols, including Ethernet/IP, FINS, Modbus, and PROFINET.

Without enough evidence of malicious intent, Dragos concluded that the tool was likely a red team tool for vulnerability scanning and probably not ICS malware. This outcome -- Not ICS malware, but a research or red team tool -- is common when hunting for ICS malware. Dragos encounters many tools that look like research, experiments, or red team tools where ICS-capability and adverse effects are clear, but malicious intent is not. Rather than assessing a sample as malware because of its potential for malicious use, the definition forces analysts to consider whether the sample has legitimate evidence of malicious intent for OT. The following case study demonstrates this process by laying out features of the IoT Exploit Tool against the three properties of ICS malware.

ICS Malware Property	Evidence
ICS-Capable	Yes, partial support for 19 industrial protocols.
Malicious Intent	No, insufficient evidence, and too many dual-use dilemmas.
Adverse Effects on OT	Yes, the use of public POCs (Proof of Concept) for public OT vulnerabilities with known negative effects.
ICS Malware? Probably not; likely a red team tool for vulnerability scanning.	

ICS-Capable: Yes

The IoT Exploit Tool is ICS-capable, with roughly 175 exploits against OT devices and numerous industrial protocols. While the tool does not have complete implementations of industrial protocols, it supports them as far as it needs to deliver the corresponding exploit.

For example, IoT Exploit Tool contains an exploit for CVE-2018-7845, a Unified Messaging Application Service (UMAS) information disclosure vulnerability reported with a public proof of concept by Talos researchers in 2018.[23] UMAS is a proprietary Schneider Electric management protocol based on Modbus TCP.[24]

To take advantage of the vulnerability, the exploit handcrafts the UMAS packet. Since IoT Exploit Tool's version of the exploit is in Cython, Talos's Python source is shown below for ease of reading.

```
def send_message(sock, umas, data=None, wait_for_response=True):
    if data == None:
        packet = ModbusADURequest(transId=1)/umas
    else:
        packet = ModbusADURequest(transId=1)/umas/data
    msg = "%s" % Raw(packet)
    resp = ""
    sock.send(msg)
    if wait_for_response:
        resp = sock.recv(2048)
    return resp

def main():
    rhost = "192.168.10.1"
    rport = 502
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(10)
    s.connect((rhost, rport))

    try:
        print "[!] This will take a few minutes"
        mbtcp_fnc = "\x5a"
        session   = "\x00"
        umas_fnc  = "\x20"
        unknown1  = "\x00"
        block_num = "\xd4\x8c"
        size      = "\xff\x00"

        outfile = "modicon_m580_dump.bin"
        with open(outfile, 'wb+') as f:
            print "[*] Writing data to %s" % outfile

        # loop through all possible offsets, writing output to file
        for i in xrange(0x19668, 0xffffffff, 0xff):
            offset = struct.pack("<I", i)
            umas = "%s%s%s%s%s%s%s" % (mbtcp_fnc, session, umas_fnc, unknown1, block_num, offset, size)
            ret = send_message(s, umas)

            with open(outfile, 'ab') as f:
                f.write(ret[13:])
```

Pack UMAS data into Modbus Packet and send

Hard-coded UMAS info

Build UMAS data

Build and send UMAS request

CVE-2018-7845 Exploit Code from Talos

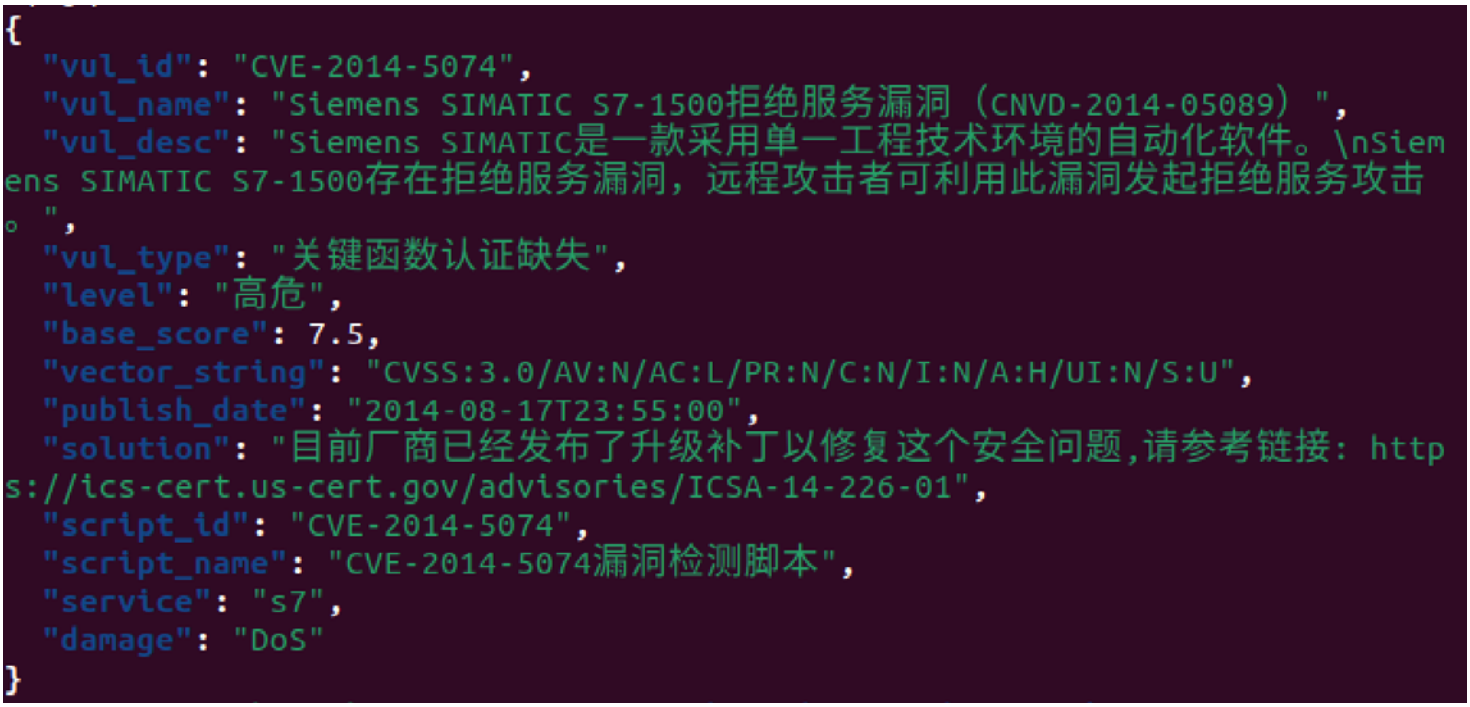
The exploit relies on a Scapy Modbus TCP implementation but is not a full-featured UMAS implementation. Even though its level of interactivity is limited, it still effectively communicates with the PLC, so this exploit is ICS-capable.

The same logic applies to all the OT exploits in IoT Exploit Tool's catalog of CVEs. While the tool demonstrates other examples of ICS-capability, the exploit catalog is more than sufficient evidence for this property.

Malicious Intent: No

The code appears professionally developed. It contains documentation describing many components and their usage, as well as a JSON file with detailed vulnerability information such as the vulnerability name, CVSS string, vulnerability ID source information, publication date, and classification. It is not apparent how useful this information would be for an offensive operation. The effects of an exploit are far more important (i.e., privilege escalation) than where the vulnerability is publicly listed and the date it was published. However, from a defensive standpoint, vulnerability information would help to develop a mitigation plan for affected devices.

23 TALOS-2018-0745 - Talos Vulnerability Report
24 The secrets of Schneider Electric's UMAS protocol - Kaspersky ICS CERT



Excerpt of IoT Exploit Tools JSON file showing CVE-2014-5074

IoT Exploit Tool's Vulnerability Classifications

- All vulnerability detection scripts
- Non-destructive vulnerability detection scripts
- All video equipment vulnerability detection scripts
- Non-destructive vulnerability detection scripts for video equipment
- All industrial control equipment vulnerability detection scripts
- Industrial control equipment non-destructive vulnerability detection scripts
- All routing and switching equipment vulnerability detection scripts
- Routing and switching equipment non-destructive vulnerability detection scripts
- All office automation equipment vulnerability detection scripts
- Office automation equipment non-destructive vulnerability detection scripts
- Windows vulnerability detection scripts
- SMB service vulnerability detection scripts
- Windows configuration vulnerability authorization detection scripts
- Browser vulnerability detection scripts
- Mail service vulnerability detection scripts
- FTP service vulnerability detection scripts
- DNS service vulnerability detection scripts
- RPC service vulnerability detection scripts
- SNMP service vulnerability detection scripts
- Trojan detection scripts
- PLC vulnerability detection scripts
- SCADA vulnerability detection scripts
- RTU vulnerability detection scripts
- Industrial control programming software vulnerability detection scripts

IoT Exploit Tool contains many instances of a dual-use dilemma — features for which both offensive and defensive explanations can be found. For example, it includes the ability to detect a limited set of malware such as Mirai and DarkComet. Virus detection is usually considered a defensive capability, but an adversary may want to detect the presence of other attackers on a target. IoT Exploit Tool features simple obfuscation through Cython and another tool for replacing module and file names with randomly generated ones. This type of obfuscation is used by both malware developers to hinder malware analysts and regular software developers to protect intellectual property. Likewise, the bundled exploits and vulnerability scanning could be used by an adversary for intrusion or by a red team to test defenses.

Nothing in the binary analysis provides conclusive evidence the tool was designed maliciously for OT environments. All the analyzed OT exploits were for publicly disclosed vulnerabilities, and there was no evidence of 0-day development. Further, with no evidence of the tool's use, Dragos can only theorize as to its purpose for which there are defensive and offensive cases. Dragos found persona information that led to a likely organization behind the tool. Still, after an investigation of that organization, the tool is as likely to be a red-team tool as an offensive cyberweapon. Lastly, Dragos had no outside intelligence about the tool from partners. For these reasons, Dragos concluded there was insufficient evidence of malicious intent.[25]

Ability for Adverse Effects on OT Environments: Yes

The catalog of CVEs and exploits is evidence for the adverse effects property. Each exploit can achieve a corresponding adverse effect. For example, the tool takes advantage of CVE-2018-5459, a CODESYS runtime vulnerability in WAGO PFC200 series devices.[26] When exploited, this vulnerability provides unauthorized access to the PLC, allows arbitrary read, write, and delete access, and the manipulation of running programs.[27] Unauthorized access and program manipulation of a PLC are both adverse effects for the OT environment.

Dragos verified that IoT Exploit Tool uses publicly disclosed CVEs by comparing the tool's modules to the corresponding public POCs. Since those POCs and the CVEs were verified by third parties and, in some cases, discovered by Dragos researchers, like the previous CODESYS vulnerability, the catalog of exploit modules satisfied Dragos's threshold for the "ability" requirement of the adverse effects property.

While Dragos analysts were able to build and install the tool, it uses an unidentified protection library that requires a run key to execute the modules. At that point, there was not enough evidence that the tool was malicious to merit spending time bypassing the library. However, the protection library did not obfuscate the underlying assembly or strings, making static analysis possible.

The ability to investigate the underlying assembly and compare it to public POCs revealed enough evidence to conclude that the tool has the ability for adverse effects on OT environments.[28]

Result

IoT Exploit Tool is probably not ICS malware. Since it partially supports 19 industrial protocols, it is ICS-capable. Since Dragos could verify its use of public POCs for known OT vulnerabilities, it has the ability to cause adverse effects on OT environments. However, there was not enough evidence of malicious intent. It only has two ICS malware properties, so IoT Exploit Tool is assessed as a likely red team tool for vulnerability scanning.[29]

Whether IoT Exploit tool is malware, the fact remains that it could be abused for harm against OT. For this reason, Dragos analysts wrote detections for it and continue to hunt for its use and new versions.

25 From an intelligence standpoint, Dragos reached a low-confidence assessment of malicious intent for IoT Exploit Tool.
26 ICSA-18-044-01 WAGO PFC200 Series - CISA
27 Critical CODESYS Vulnerabilities in WAGO PFC 200 Series
28 In an intelligence context, analysts weighed the inability to run the tool against the tool's verified use of public POCs for known CVEs and made a moderate confidence assessment in the tool's ability for adverse effects.
29 From an assessment perspective, IoT Exploit Tool is potential ICS malware, malware for which a small change in evidence could cause us to assess it as ICS malware. For example, evidence from a trusted intel partner linking the tool to a threat group could shift the sample into the ICS malware category.

IOControl

In the Fall of 2024, Dragos discovered IOControl as part of a widespread campaign by the threat group BAUXITE.[30] During this campaign, which ran from late 2023 into 2024, BAUXITE deployed IOControl against 400+ internet-exposed OT/IoT devices and firewalls. On the OT side, victim devices were made by vendors such as Phoenix Contact, Red Lion Controls, Unitronics, Tridium, Baicells, and Orpak.[31]

IOControl is an embedded Linux backdoor. Dragos analyzed two samples of it, which were discovered on two separate ARM devices manufactured by Orpak and Phoenix Contact. The malware provides the attacker with access to the victim's device for the following actions:

- exfiltrate system and user data
- run Linux commands
- conduct port scanning
- wipe device memory and storage media

The malware uses Message Queueing Telemetry Transport (MQTT) port TCP/8883 to communicate with a command-and-control (C2) server.[32] But, it does not use MQTT to interact with OT or IoT devices. After analysis, Dragos concluded the sample shows evidence of malicious intent and the ability for adverse effects, but it is not ICS-capable. Thus, IOControl is not ICS malware. Instead, Dragos refers to it as an embedded Linux backdoor targeting IoT and OT devices.

Since OT networks include devices based on operating systems like Windows and Linux, they can be affected by IT or IoT/embedded malware. By applying the ICS malware definition, analysts must consider whether a sample's abilities and interactions require an OT environment to be effective. As shown here with IOControl, this process helps analysts distinguish between IT and IoT malware that happens to operate in an OT environment and ICS malware that is truly ICS-capable.

ICS Malware Property	Evidence
ICS-Capable	No. It only has general embedded Linux functionality, and MQTT is used for C2, not OT interaction.
Malicious Intent	Yes. Evidence of targeting OT devices and association with BAUXITE.
Adverse Effects on OT	Yes. Unauthorized access to OT and more.
ICS Malware? No	

ICS-Capable: No

IOControl is not ICS-capable. From the malware's functions, Dragos ruled out three features that may initially appear to be ICS capabilities.

30 BAUXITE - Dragos
31 Page 26 - Dragos 2025 OT Cybersecurity Report
32 MQTT.org

Runs Linux Commands

IOControl can execute arbitrary commands via the **system** API call[33] or directly using a shell like Bash.[34] This type of ability can have an effect on ICS, but the code for executing commands is not specialized for an ICS environment. The function would work on any Linux system, because it codifies general Linux knowledge not ICS/OT knowledge. A helpful question for distinguishing these cases: "Does this code require an OT environment to be effective?" If the answer is no, like in this case, it is likely not an ICS capability.[35]

Wipes device memory and storage media

One version of IOControl featured a (Memory Technology Device) MTD wiper, which uses proc/mtd to find flash memory device files and erase the blocks within. This type of memory is associated with embedded devices and can be found in OT and IoT devices. However, the code for wiping embedded devices is not specialized to an ICS/OT environment. It would work on any embedded Linux device, whether a handheld gaming console, an IP camera, or an ICS PLC. For this reason, an MTD wipe is not an ICS capability but a more general embedded Linux one.[36]

Uses MQTT for Command and Control

MQTT is traditionally an IoT protocol that can be found in ICS devices. If the malware used MQTT to send commands to PLCs in the OT network, this would be an ICS capability. However, the protocol is only used to speak to the attacker's C2 server, so it does not qualify in this case.[37]

Since there are no qualifying ICS-capable functions, IOControl is not ICS-capable malware.

Malicious Intent: Yes

IOControl shows evidence of malicious intent in its use against OT devices and its association with the BAUXITE threat group, which has strong technical overlaps with CyberAv3ngers[38] and a history of targeting OT systems.[39] BAUXITE used IOControl to target OT devices made by Orpak, Phoenix Contact, Red Lion Controls, Unitronics, and others. In October of 2023, in a public display of malicious intent, CyberAv3ngers boasted on Telegram about attacking gas stations in Israel by compromising Orpak systems. They released screenshots of a pump station control panel and the output of a wiper tool allegedly attacking Orpak devices, along with other information about the victims.[40] Dragos later learned that BAUXITE had deployed IOControl to a fleet of Orpak fuel management devices, but an error in the malware installation caused the devices to go offline.[41] Despite the accident, the intent behind IOControl was obvious and malicious.

Ability for Adverse Effects on OT Environments: Yes

While IOControl is not ICS-capable, it can have several adverse effects on OT. Here are a few:

- A remote backdoor provides unauthorized access to the OT network.
- The wiper ability can damage and take the devices off-line.
- The data exfiltration function and arbitrary Linux command ability can steal information.

33 system(3) - Linux manual page
34 Bash (Unix Shell) - Wikipedia
35 Dragos applied the same logic to ransomware cases like EKANS, which terminated ICS-related processes. Windows process termination is not an ICS function but a standard Windows one.
36 Dragos used the same logic when assessing AcidPour since it features a similar ability to wipe MTD flash memory and SD/MMC cards.
37 Protocols that can be found in any environment (IT, IoT, or OT) require an evaluation of use. In the PIPEDREAM framework, BADOMEN uses HTTP to communicate with Omron PLC CGI endpoints to manipulate the target devices. In that case, Dragos considered the communication code to be an ICS capability. Reference: Analyzing PIPEDREAM: Results from Runtime Testing - Dragos
38 https://attack.mitre.org/groups/G1027/
39 Page 22 - Dragos 2025 OT Cybersecurity Report
40 CyberAveng3rs stopped using their Telegram channel, but Team82 captured screenshots in their public analysis here: Inside a New OT/IoT Cyberweapon: IOCONTROL
41 An installation error could potentially violate the adverse effects property, but Dragos had evidence of successful installation on many other devices.

```
Parse_Data_For_Actions(local_1c,auStack4064,aiStack8064);
iStack12072 = 0;
for (iStack12068 = 0; (iStack12068 < 1000 && (aiStack8064[iStack12068] != 0));
    iStack12068 = iStack12068 + 1) {
  __s = (char *)memcpy(aiStack8064[iStack12068],&DAT_0000f1d4);
  switch(auStack4064[iStack12068]) {
  case 0:
    Gather_And_Send_Host_Data(local_18);
    break;
  case 1:
    Check_Binary_Permissions(local_10,__s);
    break;
  case 2:
    iVar1 = Run_Cmd_Received_From_C2(__s);
    aiStack12064[iStack12068] = iVar1;
    break;
  case 3:
    Self_Delete(local_18);
    break;
  case 8:
    local_20 = strtok(__s," ");
    while (local_20 != (char *)0x0) {
      local_34[iStack12072] = local_20;
      local_20 = strtok((char *)0x0," ");
      iStack12072 = iStack12072 + 1;
    }
    iVar1 = atoi(local_34[2]);
    iVar1 = Port_Scan_Range_of_IPs(local_34[0],local_34[1],iVar1);
    aiStack12064[iStack12060] = iVar1;
    iVar1 = memcpy(aiStack12064[iStack12060],local_34[2]);
    aiStack12064[iStack12068] = iVar1;
  }
```

Decompilation of IOControl's Command Dispatch

Although Dragos could not acquire the right equipment to test IOControl in the lab, Dragos analysts investigated the code using static analysis and emulation, which provided evidence that the tool worked. Further, Dragos had high-confidence evidence of successful IOControl infections across 400 devices, including OT devices. Given the ability to identify several adverse effects and verify the code worked, Dragos concluded IOControl has the ability for adverse effects on OT environments.

Result

IOControl is not ICS malware because it only has two ICS malware properties. It shows evidence of malicious intent through its targeting of OT devices and association with BAUXITE. Dragos could identify several adverse effects and had evidence of 400+ successful infections that included OT devices, so IOControl has the ability for adverse effects on OT environments. However, its capabilities showed no evidence of ICS specialization. They were either too general or, as was the case with its use of MQTT, not used to interact with devices in the OT environment. For this reason, IOControl is not ICS-capable.

Conclusion

The ICS Malware Definition is a new tool for identifying and analyzing malware targeting ICS. By applying the ICS Malware properties—ICS-Capable, Malicious Intent, and Adverse Effects—this definition differentiates ICS Malware from red team tools, standard software, and other malware that target ICS but lack ICS-specific capabilities. For the ICS community, the definition provides a consistent language for discussing aspects of ICS malware and a method for evaluating new malware released in reports: “What are its ICS capabilities? Do I have those affected technologies in my environment?”

The ICS Malware Definition is based on Dragos’s collective experience with ICS malware to date. It is designed to be flexible, allowing for the subjective judgments that are often required in malware and intelligence analysis. However, attackers do not care whether their tools fit within the ICS Malware Definition or any defender framework. So, Dragos expects new strains of malware targeting OT, including ICS malware, that will challenge and expand the understanding and definition of this concept. As Dragos’s understanding continues to grow, this definition will evolve alongside it.