You Can't Protect What You Don't Understand: Characterizing an Operational Gas SCADA Network

Xi Qin*†, Martin Rosso*‡, Alvaro A. Cardenas†, Sandro Etalle‡, Jerry den Hartog‡, and Emmanuele Zambon‡
* Authors contributed equally; the remaining authors are listed in alphabetical order
† University of California, Santa Cruz. California, USA. {xqin9, alvaro.cardenas}@ucsc.edu
‡ Eindhoven University of Technology. The Netherlands. {m.j.rosso, e.zambon.n.mazzocato, j.d.hartog, s.etalle}@tue.nl

***Abstract*—Natural gas distribution networks are part of a nation's critical infrastructure, ensuring gas delivery to households and industries (e.g., power plants) with the correct chemical composition and the right conditions of pressure and temperature. Gas distribution is monitored and controlled by a Supervisory Control and Data Acquisition (SCADA) network, which provides centralized monitoring and control over the physical process.**

In this paper, we conduct the first openly available network measurement study of the SCADA network of an operational large-scale natural gas distribution network. With a total of 154 remote substations communicating through the SCADA system with a Control Room and over 98 days of observation, this is, to the best of our knowledge, the most extensive dataset of this kind analyzed to date.

By combining the information obtained from engineering and IEC 104 network traffic, we reconstruct the gas distribution system's layout, including the type and purpose of the substations and the physical properties of the gas that enters the SCADA system. Our analysis shows that it is possible to extract this information, essential for security monitoring, purely from the raw network traffic and without background knowledge provided by the control system engineers. We also note that configuration changes in SCADA environments, although probably less frequent than in IT environments, are not as rare and exceptional as the research community assumed.

***Index Terms*—SCADA, IEC 60870-5-104, gas distribution network, cyber physical systems, Telnet**

I. Introduction

Natural gas makes up "for 23% of global primary energy demand and nearly a quarter of electricity generation" [1]. The British oil and gas company BP estimates that roughly 3.8 trillion m^3 of gas were consumed in 2020 worldwide [2], and natural gas consumption is growing. Compared to other fossil fuels (e.g., coal, diesel, gasoline), natural gas is often considered a more environmentally friendly energy source; a move from coal to natural gas could potentially cut carbon emissions by half [3], [4]. In fact, most of the reduction in America's CO_2 emissions between 2005 and 2019 was because of the switch from coal to gas [5]. As natural gas becomes a more prevalent energy source, gas distribution networks become an even more essential part of our critical infrastructure. As the Colonial pipeline ransomware demonstrated, attacks to our critical energy transmission and distribution pipelines can have severe consequences (e.g., closing airports, rising costs of fuel, and panic through the population) [6].

Despite their growing importance, the academic community knows very little about how these pipeline networks are managed and operated. The (general) inaccessibility of industrial control datasets, as noted by the International Energy Agency (IEA) in [7], and absence of expert knowledge as a ground truth by researchers remains today a major obstacle in the protection of Supervisory Control and Data Acquisition (SCADA) systems. This paper presents the first network traffic measurement and analysis of a real-world gas distribution system. To the best of our knowledge, we are the first to study the SCADA network of operational gas infrastructure. Our network capture consists of almost 100 days of continuous communications between a central control station and over 150 remote substations separated by several kilometers.

We reverse-engineer and reconstruct the network by analyzing the operators' commands sent over by Telnet and IEC 60870-5-104 (IEC 104) services. We track network maintenance events and present an algorithm to re-identify Remote Terminal Units (RTUs) in a changing network using fingerprinting techniques. Our analysis shows that it is possible to extract information about gas networks purely from the raw network traffic and without background knowledge provided by the operators. We also note that configuration changes in SCADA environments, although probably less frequent than in IT environments, are not as rare and exceptional as the research community assumed.

II. Background and Related Work

Operators monitor and control the gas network with the help of a SCADA system. A typical SCADA system has at least one control center and multiple geographically distributed remote stations. In the control center, the operator can see the status of the gas network through a Human Machine Interface (HMI) and can invoke remote control commands. The control center also has several other services, such as a database with historical data and time server for the network clock.

The SCADA server communicates with Remote Terminal Units (RTU) located in (remote) substations. RTUs interact directly with the equipment in the substation., including sensors (e.g., pressure sensors), actuators (e.g., valves), as well as other Intelligent Electronic Devices (IEDs), and Programmable Logic Controllers (PLCs).

In the last decade, the communication technologies used for supervision and control of industrial control systems have migrated from serial links to IP-network protocols. In case of gas distribution, these networks are usually private

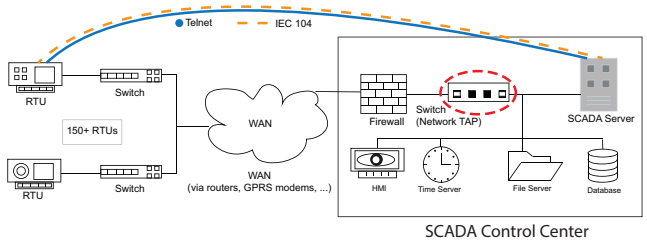

Figure 1. SCADA network for our study. A dashed oval highlights the capture location. The only application-layer protocols used between the control center and the RTUs are IEC 104 and Telnet.

networks composed of dedicated leased lines provided by a telecommunications company; Despite SCADA networks not being exposed to the public Internet, they use IP-compatible protocols such as Modbus/TCP, DNP3, and IEC 104.

Over the last ten years, research on SCADA networks has grown significantly, however, most network security research uses emulated networks, synthetic SCADA data [8]–[11] or, simple testbeds, rather than real-world operational SCADA systems. Outside of security experiments in testbeds, network measurements of SCADA networks have been inadequate. The small number of studies from real-world operational SCADA systems focus only on the power grid [12], [13]. As far as we are aware, this paper is the first analysis of a real-world operational gas SCADA network.

III. Network Capture

We obtained the dataset from the SCADA network of a European gas distribution operator. The packet capture contains traffic exchanged between devices and services within the control center (i.e., SCADA servers, HMI stations, workstations, file servers, time servers, printers, network switches, and routers) and remote substations (RTUs) several kilometers away from the control center. The remote gas substations perform the roles of gas generation, regional transmission, and distribution.

As illustrated in Figure 1, we capture data from a mirror port of the main network switch serving the control center network. This particular vantage point allows us to capture all interactions among devices in the control center network and the interactions with the remote devices (RTUs). The control room is connected to remote RTUs via GPRS and dedicated leased lines.

The recorded traffic spans 98 days of summer, with 79 days of capture being consecutive and seven days missing in the last month due to a transient fault in the port mirroring of the network switch. The overall capture time, after concatenation, is 2037 hours or 84.9 days. The dataset is around 500 GB in size and contains over 2.2 billion network packets in total, resulting in an average of 298 packets per second over 98 days.

We observe a total of 397 distinct IP addresses distributed over 23 different class-C network segments that we enumerate with letters A to W. Two IP addresses belong to mobile (phone) network operators, 361 addresses belong to the network segment assigned to the gas distribution company, and 25 addresses are from private network segments. The remaining nine are link-local IP addresses, eight of which are IPv6.

By analyzing ARP traffic, we observe that 22 hosts are in the Ethernet broadcast domain handled by the network switch from which we are capturing the traffic, while the remaining 375 hosts are behind network gateways (i.e., either routers or firewalls). By analyzing the application-layer protocols, we infer that 304 hosts on the network are RTUs, two hosts are SCADA servers, four are HMI stations, two are time servers, one is a printer, two are network switches/routers, and the remaining 82 IP addresses correspond to other workstations deployed in the control room.

Within the control room, we observe a mix of typical IT application layer protocols (DHCP, DNS, HSRP, NetBIOS, Oracle TNS, SLP, SMB, SNMP, SSDP, NTP, SSH, Web Service Discovery, X11) and a proprietary industrial protocol used both for communication between the two SCADA servers, and between the HMI workstations and the SCADA servers.

However, only two application-layer protocols are used to communicate with the gas substations. These protocols are IEC 104 (an industrial control protocol) and Telnet. Of the 2.2 billion network packets, almost 118 million (5.3%) are IEC 104, and almost 72 thousand (0.003%) are Telnet. We focus on the communications between the control center and the remote substations in the remaining sections of this paper.

IV. Data Analysis

When analyzing industrial network data, researchers often face the challenge that network, and even more importantly, process control configuration information and expert knowledge, is often unavailable or inaccurate. This unavailability usually happens because of two reasons: (a) configuration data is deemed sensitive and thus is not shared externally (e.g. to researchers); (b) information is scattered among several different groups and often not adequately maintained, making it difficult to collect and to ensure its soundness and accuracy.

Since we did not get expert knowledge from the gas network operator, we have to find data-driven ways of extracting the network structure. During our first analysis of the system, we enumerated all IP addresses that used the IEC 104 protocol. We identified two IP addresses in the control room (two SCADA servers) and 217 remote IP addresses for RTUs.

First, to track *all* IP address changes, regardless of whether they were done locally or via Telnet, we present an algorithm to uniquely identify RTUs even after their IP address changed in Section IV-A. To the best of our knowledge, we are the first to provide an algorithm tracking RTUs by their IEC 104 traffic. Then we will explain in Section IV-B and IV-C, we found Telnet sessions between the control center and a subset of these remote RTUs, and by analyzing these sessions, we noticed that some RTUs had their IP address reconfigured via Telnet. As we can observe RTUs changing IP addresses without a matching Telnet interaction, we conclude that some RTUs were reconfigured locally, i.e., by a technician at the substation.

Input: Sequence of IEC 104 messages $msg_1 \dots msg_M$
Init: $D_{ip} \leftarrow \{\}$ $D_{ioa} \leftarrow \{\}$ $D_{\Delta ip} \leftarrow \{\}$
For $i = 1$ to M
  $t_i, ip_i, \{(a_{i1}, o_{i1}), \dots (a_{in}, o_{in})\} = parse(msg_i)$;
  $rtu = a_{i1}$;
  AddSet $\{o_{i1}, \dots, o_{in}\}$ to $D_{ioa}[rtu]$;
  If $ip_i \neq D_{ip}[rtu]$ then
    AddElement $(D_{ip}[rtu], ip_i, t_i)$ to $D_{\Delta ip}[rtu]$;
    $D_{ip}[rtu] \leftarrow ip_i$
  EndIf
EndFor

Algorithm 1: RTU fingerprinter & IP change detector

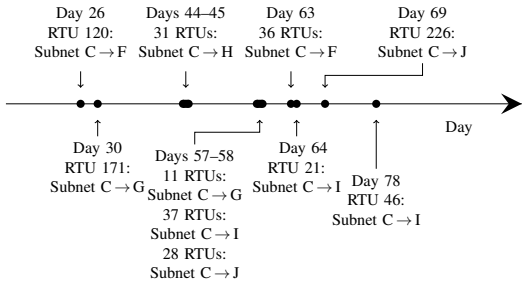

Figure 2. Timeline of IP reconfiguration activities in the IEC 104 network

A. Tracking Unique RTUs by their IEC 104 features

According to the IEC 101/104 protocol specification [14], [15], the protocol defines two primitives, Common Address (CA) and the Information Object Address (IOA), which together uniquely identify a physical process variable in the system. Only I-format IEC 104 messages encapsulate these two primitives. An I-format IEC 104 message is for transmitting the process variables' values within one or more Application Service Data Unit (ASDUs). Each ASDU provides a CA and one or more information objects that each incorporates an address, i.e. IOA. Therefore, CA can be understood as a local post station for the monitor and control variables utilized in the physical process, and IOA resembles a local address on a device employed in the physical process.

An ASDU message links an IP address to the specific RTU, managing unique process variables, identified by a set of tuples of CA and IOA. Then, any new IP address linked to this known CA-IOA pair implies an IP change of the RTU. As illustrated in Algorithm 1, we catalogue all CA-IOA tuples and cross-reference them with source IP address, to identify all IP address changes of RTUs.

In our dataset we observe that CAs are unique across all RTUs, allowing us to use them as RTU identifiers ($RTU = CA$). The algorithm outputs two dictionaries. $D_{ioa} : RTU \rightarrow \mathcal{P}(IOA)$ maps which IOAs belong to which RTU, and $D_{\Delta ip} : RTU \rightarrow \mathcal{P}(IP \times IP \times TimeStamp)$ records derived IP changes (from old to new ip, derived at given time). Given a method $parse : Msg \rightarrow TimeStamp \times IP \times \mathcal{P}(CA \times IOA)$ that extracts the RTU's IP and associated (CA, IOA) pairs from an IEC 104 message. The algorithm uses $D_{ip} : RTU \rightarrow IP$ to store the last known IP of each RTU.

1) Observations and Results: Applying the algorithm we conclude there are 154 unique RTUs in this gas distribution network. IP reconfiguration events are found on several days as shown in Figure 2. The last occurrence in our capture was on day 78. Based on our observations in Telnet sessions, we further testify some of the configuration events, one IP address change on day 30 and another 17 on day 44 (see Section IV-B). Manual investigation reveals that the IP address reconfigured on day 22 was assigned to an RTU that was still under configuration and did not show any IEC 104 traffic before day 22. The algorithm finds additional IP address reassignments in which two initial subnets C and I were migrated to four subnets F, G, H, and J.

We have thus obtained the total number of unique RTUs in the dataset and how they change over the course of the summer. As illustrated in Figure 3, at the beginning of our network capture, a subset of RTUs got connected to one control server, and then the whole network was reconfigured to connect to a different control server. During our 100 days, we see a significant change in the SCADA network, and we believe such a significant network change was scheduled during the summer (the dates of our capture), the period of low demand for gas (residential consumers use gas for heating) when potential outages have less severe consequences. This network reconfiguration is also an indication that IP addresses, even in SCADA networks, should not be modeled as an invariant and that SCADA networks in general are not as static as previously suggested by the intrusion detection community [16]–[19].

B. Analyzing Telnet Sessions

While searching the network capture for the presence of engineering and remote management protocols, we find, besides the IEC 104 traffic, 878 Telnet sessions between the SCADA servers and the RTUs. In this subsection, we analyze these Telnet sessions searching for configuration information and changes in the network or process control configuration. The *clear text* nature of the Telnet protocol allows us to reconstruct a ground-truth of the state and configuration changes for all RTUs in the network that received Telnet engineering traffic.

We first extract and (re)assemble all Telnet sessions, using the `chaosreader` program [20]. Besides the Telnet commands, `chaosreader` extracts the timestamp of the first packet and the duration (i.e., the time difference between the first and the last seen Telnet packet) for each session.

One particular aspect of interest is whether an operator manually initiated telnet sessions or the sessions were part of automatic (scheduled) machine-to-machine interaction. This will help us understand whether the management traffic is part of ongoing maintenance or scheduled diagnostic traffic. We also look for indicators of human-handled Telnet sessions, such as the time of day the session was initiated at, its duration, the time between keystrokes during the session, and typos made during command and credential transmission.

In total, we observe Telnet sessions on 34 non-consecutive days. For 64 days we do not observe any Telnet traffic at all. We visualize the distribution of Telnet sessions over the whole dataset in Figure 4. 94% (828/878) of the Telnet sessions occur between days 9 and 44. In this time span, 71% (625/878) of

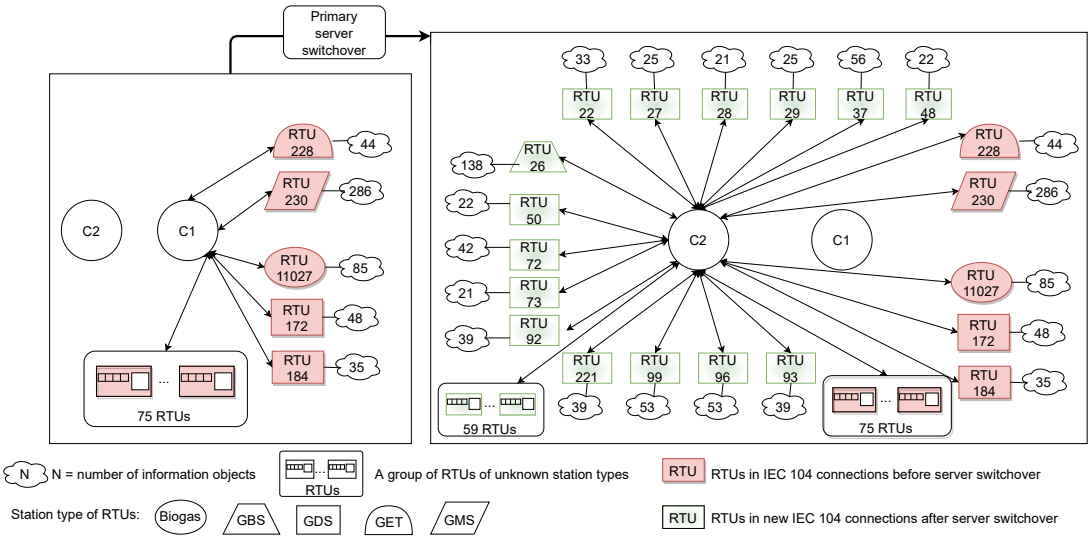

Figure 3. IEC 104 Network Topology before and after switching SCADA Servers

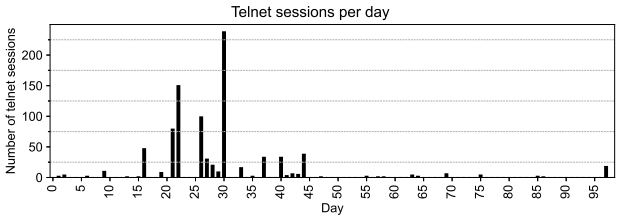

Figure 4. Number of Telnet sessions per day

the sessions occur within the ten days between day 21 and 30. The high amount of Telnet sessions in a relatively short amount of days is probably caused by ongoing maintenance. Looking at the time distribution of Telnet sessions, we only observe sessions on weekdays.

The majority of Telnet sessions lasts for eight seconds or less. A manual investigation of 152 sessions shorter than four seconds reveals that even though these sessions were fully initiated, the initiator never provides credentials to sign-in on the RTU. Further, the majority of short sessions for which the session initiator actually authenticated, show a super-human typing speed of more than 50 characters within a fraction of a second. In all short sessions only a single custom command, `gsmsms`, got executed, with a timestamp and an informational log or error message as arguments. The typing speed and execution of the same basic command are strong indicators for scripted Telnet interactions executed by a machine. However, the fact that all these sessions occurred during working hours indicates that the automated process was triggered by a human.

The longer Telnet sessions appear to be fully manual. We find typos and wrong commands, e.g., "`setop`" (sic) or "`ip rout add`" (sic) and wrong command arguments, e.g., "`route add default enet0 <ip address>`" which is later corrected to "`route add default <ip address>`". We also find 198 occurrences of the backspace character indicating the correction of a typo (i.e., there is no obvious reason for a machine to use the backspace character as machines do not make typos). This further enforces our conclusion of most complex sessions being actively carried out by humans as part of the maintenance activity.

Besides the two SCADA servers initiating Telnet sessions, we identify a third IP address that tries to initiate TCP sessions on destination port 23 (i.e., Telnet) five times in total within a period of nine days between days 13 and 21 inclusive. The originating IP address belongs to RTU 49 and the target is one of the SCADA servers. The SCADA server resets the TCP sessions. One possible explanation for the connection attempts is that an operator managing the RTU on-site by connecting to its serial port tried to access the SCADA server to retrieve some information needed to carry out the maintenance job, instead of connecting to the SCADA server directly from her computer. This hypothesis is supported by Microsoft Browser service messages from a new device being captured during the same days the connection attempts were recorded. The device identifies itself as a Windows XP machine with host comment "`Configuration PC Datawatt NIV`".

C. Understanding RTU Details in Telnet Sessions

We analyze the extracted commands with a special focus on (1) network configuration and (2) physical process commands. Using a regular expression, we count 211 Telnet sessions in which at least one IP address is present in the payload. More precisely, we find IP addresses are used as parameters for two types of network configuration commands: (1) `ifconfig enet0 <IP address> [netmask <subnet mask>]`, which sets the IP address of the local device and (2) `route add default <ip address>` which sets the device's default gateway.

In most cases the RTU is assigned the IP address already in use, thus the command does not have any effect. On day 22, a device from subnet C was reconfigured to listen on a

```
===>> Software revision [REDACTED] <<==
RTU [REDACTED] Type=[REDACTED] D05_V4 Name='[REDACTED]'
[...]

Ai-01 Name='[REDACTED]' Min=-40 Max=160 Dim='millibar'

Di-41 Name='[REDACTED] LL' On='DEFECT' Off='OK'
Di-42 Name='[REDACTED]  L' On='ALARM'  Off='Normal'
Di-43 Name='[REDACTED]  H' On='ALARM'  Off='Normal'
Di-44 Name='[REDACTED] HH' On='ALARM'  Off='Normal'

Ao-01 Name='[REDACTED] LL' Min=-40 Max=160 Dim='millibar'
Ao-02 Name='[REDACTED]  L' Min=-40 Max=160 Dim='millibar'
Ao-03 Name='[REDACTED]  H' Min=-40 Max=160 Dim='millibar'
Ao-04 Name='[REDACTED] HH' Min=-40 Max=160 Dim='millibar'
```

Figure 5. Simplified, output of the `prnrtu` command

new IP address from the same subnet multiple times within 10 minutes. On day 30 one RTU was moved from subnet C to subnet G. Finally, on day 44, 17 RTUs from subnet C were moved to subnet H within a time frame of roughly 1.5 hours.

These observations indicate that during data capture there was ongoing network maintenance, in which multiple RTUs were moved from one subnet to another. We have reason to believe that most Telnet sessions are part of a previously scripted and semi-automated maintenance process, which would explain why some devices received IP address reconfiguration commands that did not result in any changes. [1]

Of all commands we observe within the Telnet sessions, the vendor-specific shell command `prnrtu` is of special interest. As the name suggests, the command prints on screen a variety of information about the RTU; a simplified output of this command is reported in Figure 5. Thanks to this command, we are able to obtain the model of the RTU, the type of substation the RTU is monitoring/controlling, and the specific variables and alarm configurations in the substation.

Model: All RTUs are Datawatt D05, with a variety of different software versions. Datawatt is a Dutch manufacturer and the D05 is a modular RTU designed to be deployed in different kinds of substations, supporting both IEC 101 (serial) and IEC 104 (TCP) communications.

Type of substation: Because of the way RTUs are named, we are able to infer the role of the station within the gas distribution network, as well as its geographical location within the operator's distribution area. Table I shows the different types of substations we identified.

Process Variables: Variable configuration is composed of an identifier, followed by a number of additional parameters. The identifier is constructed from a prefix D/A (for binary or analog), i/o (for input or output), followed by a simple counter. The first parameter is the variable name, which provides a semantic description of what values the variable stores. For binary variables, there are two parameters that describe the meaning of the On and Off state of the variable in textual form. For analog variables, the possible Min(imum) and Max(imum) values are defined and a Dim(ension) parameter reports the measurement unit.

[1]We believe the script iterates over a given set of RTUs and performs maintenance actions for each one. The commands are idempotent, resulting in no effective change for RTUs that are already in the desired state.

Table I
IDENTIFIED RTU STATION TYPES (FROM TELNET)

#	Name	Description
18	Distribution Station (GDS)	Where the low pressure local transport network becomes the last mile local distribution network
1	Measuring Station (GMS)	Measures pressure variables from surrounding locations/streets.
1	Expansion Station (GET)	Where the the local distribution network connects to the the high-pressure gas transport network. To our understanding, the high-pressure gas is expanded, i.e., to reduce the pressure to the operational values of the distribution network.
1	Biogas Generator Station (Biogas)	A third party company that produces and injects biogas into the network.
1	Testing Station (GBS)	Dedicated point for testing & measuring for gas leakage and over-pressure conditions in the distribution network, with safety outlet valves to reduce pressure in case of over-pressure.

Table II
PRESENT MEASUREMENT UNITS

Roughly a quarter of these are variables and 3/4 are alarm thresholds. Various spellings (e.g., of the word "millibar") were aggregated.

Unit	Concept	Count
millibar	pressure	341
%	relative	111
bar	pressure	67
Cts	valve position	61
m^3/h	flow	11
s	time (duration)	10
°C	temperature	21
ppm	parts per million	5

Using a regular expression, we extract all configured measurement units (or "dimensions") from the Telnet configuration sessions in Table II. Pressure is measured either in bar or millibar. While the local distribution network mainly operates at a pressure of a few bar, the gas service line into private homes (i.e., the last 1/2 mile) is operated at pressure levels below 1 bar. Gas flow is measured in volume over time, i.e., m^3/h. Percent (%) and "Counts" (Cts) represent valve position (from open to closed), time duration is measured in seconds (s), and gas temperature is measured in degrees Celsius (°C). The Biogas Generation station is the only station measuring additional gas quality metrics, namely H_2O in ppm (i.e., parts per million) and O_2, CO_2, and CH_4 in percent (%).

Alerts: For each analog sensor variable, we find four alerts configured: High-High (HH), High (H), Low (L), and Low-Low (LL). These alerts are typical in process control. When a sensor value reaches an H or L value, this represents an anomaly and potential concern. When a sensor value reaches HH or LL it means it has reached the highest or lowest tolerable value and actions must be taken to bring the variable within normal ranges. For example, Figure 5 shows an analog input `Ai-01` with range -40 to 160 millibar, alarm signals `Di-41` to `Di-44`, and corresponding alarm limits `Ao-01` to `Ao-04` for LL, L, H, and HH alarms respectively. Overall,

Table III
PHYSICAL SEMANTICS OF PRESENT 1048 POINT VARIABLES

Type	Description	Station	Signal	#
Alarms	Alarm status	All stations	Binary	471
	Alarm configuration	All stations	Analog	253
Control	Regulator control	GDS, GBS	Binary	42
	Emergency control for gas expansion	GBS, GET	Binay	3
	Gas flow rate control	Biogas	Analog	1
Sensor values	Valve indicators	GBS, GDS	Binary	87
	Position indicators	GDS	Binary	65
	Pressure indicators	All stations	Analog	42
	Regulator indicators	GDS, GBS	Binary	20
	Temperature indicators	Biogas, GBS, GET	Analog	18
	Membrane indicators	GDS	Binary	13
	Flow rate indicators	Biogas, GBS	Analog	8
	Motor running indicator	GDS	Binary	7
Other	Test signal	GDS, GET, GMS	Binary	18

the Telnet traffic reveals configuration of 1048 point variables, presented in Table III.

The vast majority of these are related to alarm configuration and alarm flags, e.g., for gas pressure, flow rates, temperatures, and valves. Only a minority of point variables are actual sensor readings, e.g., for gas pressure and valve or motor positions. We believe that all this semantic information will be highly valuable for process-based intrusion detection systems, which require knowledge about variables and their relationships.

V. DISCUSSION AND CONCLUSIONS

One of the lessons learned is that while security researchers might not get much information about the process, we should **learn to utilize the management and diagnostic traffic in SCADA networks** and not rely solely on SCADA protocols which are only meant for monitoring and control. Every process control device has a diagnostic and configuration interface, usually a TCP/IP interface. In the environment we analyzed, homogeneously built with only Datawatt D05 RTUs, this interface was Telnet; in other SCADA environments it might be a proprietary ad-hoc protocol and thus require reverse engineering to achieve a complete view. We believe these protocols deserve the attention of the research community. With information from such protocols, researchers can replicate the analysis process described in this paper in other SCADA/industrial control environments.

As shown by our analysis, (network) maintenance sessions in SCADA networks are not as rare as the research community has assumed them to be: during the observed time window, we could identify both network and process control configuration traffic. In other environments, it might also be possible to leverage regular diagnostic traffic for configuration data extraction. For instance, an operator could schedule jobs that poll the device status and configuration periodically for diagnostic and integrity checking purposes. Examples of control systems supporting diagnostic polling include the most popular Distributed Control Systems such as Yokogawa Centum VP and Emerson DeltaV, and solutions typically adopted for gas pipelines such as Emerson ROC and Bristol Babcock Control Wave.

We also note that while monitoring the traffic from the vantage point of the control center gave us broad visibility into the network, multiple RTU configuration operations we inferred by inspecting IEC 104 traffic were not visible in the Telnet traffic. It likely happened because operators connected to the RTU locally, i.e., from the remote station rather than the control center. **This lack of total network visibility might be a challenge as we develop and deploy network security monitors for real-world SCADA systems.** In principle, one can solve this lack of visibility by deploying network probes at every substation, but in practice, this more expensive setup is not viable for most organizations. Instead, our approach to infer these network changes through IEC 104 traffic indicates that it is possible to identify configuration changes even when the actual (engineering) commands cannot be seen, as shown from the case study of IP reconfiguration.

The network we study is not part of the Internet (it is a private network on leased lines). However, anyone with access to this private network will have open access to all devices in the network. There is neither authentication nor encryption in the network traffic, and attackers can potentially spoof any device. Therefore, **our analysis also shows this SCADA network uses a trusted insider assumption.**

The usage of clear text protocols (i.e., Telnet and IEC 104) is (still) common practice in SCADA networks. The International Electrotechnical Commission (IEC) published a security specification for IEC 104 in 2013 to provide sender authentication and ensure the integrity of data units (i.e., APDUs). However, operators tend to be reluctant to upgrade IEC 104 channels with this security feature, probably under consideration for the expenses and interruption to routine operations. Besides most SCADA protocols predating modern best practices in protocol design, according to Fauri, Wijs, Hartog, *et al.* introducing encryption may decrease compatibility, introspection, and monitorability of the network[2], as well as introduce additional complexity and latency in the control process [21]. In this context, we do not consider plain text protocols in a private and controlled network a security issue by itself, but do caution over-optimistic trusted insider assumptions prevalent in SCADA networks. Any Man-in-the-Middle attacker could, with or without credentials gained from parsing Telnet traffic, cause significant interruptions on the gas delivery process.

Finally, we show how we reconstruct the gas distribution system's layout by combining the information obtained from engineering (i.e., Telnet) and SCADA (i.e., IEC 104) network traffic, including type and purpose of each substation and the physical properties of the gas that enters the system. **Our analysis shows that it is possible to extract this information, essential for security monitoring, purely from raw network traffic and without background knowledge provided by control system engineers. We also note that configuration changes in SCADA environments, although probably less frequent than in IT environments, are not as rare and exceptional as the research community assumed.**

[2]In fact, this work relies on data extracted from (clear text) Telnet sessions

Acknowledgment

This research was jointly funded through the international DEPICT project (grant no. 628.001.032) by the Dutch Research Council (NWO) and by the Air Force Research Laboratory and DHS under agreement number FA8750-19-2-0010. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. This research was also partially supported by NSF NeTS CNS-1929406 and NWO INTERSECT (NWA.1160.18.301). The views and conclusions contained herein are those of the authors and should not be interpreted necessarily representing the official policies or endorsements, either expressed or implied, of the Department of Homeland Security, the Air Force, or the US Government.

References

[1] International Energy Agency (IEA). "Gas – fuels & technologies." (), [Online]. Available: https://www.iea.org/fuels-and-technologies/gas (visited on 03/21/2021).

[2] BP plc, "Statistical Review of World Energy 2021," Tech. Rep., 2021, p. 72. [Online]. Available: https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2021-full-report.pdf (visited on 03/21/2021).

[3] F. Zakaria, "Opinion: We're headed for a global energy crisis. what we need is a transition strategy," Oct. 21, 2021. [Online]. Available: https://www.washingtonpost.com/opinions/2021/10/21/were-headed-global-energy-crisis-what-we-need-is-transition-strategy/ (visited on 03/03/2022).

[4] U.S. Energy Information Administration, *Frequently Asked Questions (FAQs), How much carbon dioxide is produced when different fuels are burned?* [Online]. Available: https://www.eia.gov/tools/faqs/faq.php?id=73 (visited on 03/03/2022).

[5] ——, "U.S. energy-related carbon dioxide emissions, 2019," U.S. Energy Information Administration, Sep. 2020. [Online]. Available: https://www.eia.gov/environment/emissions/carbon/archive/2019/ (visited on 03/03/2022).

[6] W. Englund and E. Nakashima, "Panic buying strikes southeastern united states as shuttered pipeline resumes operations," May 21, 2021. [Online]. Available: https://www.washingtonpost.com/business/2021/05/12/gas-shortage-colonial-pipeline-live-updates/ (visited on 03/03/2022).

[7] C. Feng, T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and lstm networks," in *47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Jun. 2017. DOI: 10.1109/DSN.2017.34.

[8] E. Hodo, S. Grebeniuk, H. Ruotsalainen, and P. Tavolato, "Anomaly detection for simulated iec-60870-5-104 trafiic," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017. DOI: 10.1145/3098954.3103166.

[9] Y. Yang, K. McLaughlin, S. Sezer, Y. B. Yuan, and W. Huang, "Stateful intrusion detection for iec 60870-5-104 scada security," in *2014 IEEE PES General Meeting — Conference Exposition*, Jul. 2014. DOI: 10.1109/PESGM.2014.6939218.

[10] P. Maynard, K. McLaughlin, and B. Haberler, "Towards understanding man-in-the-middle attacks on IEC 60870-5-104 SCADA networks," in *2nd International Symposium for ICS & SCADA Cyber Security Research*, Sep. 2014. DOI: 10.14236/ewic/ICSCSR2014.5.

[11] A. Baiocco and S. D. Wolthusen, "Indirect synchronisation vulnerabilities in the iec 60870-5-104 standard," in *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe*, Dec. 2018. DOI: 10.1109/ISGTEurope.2018.8571604.

[12] K. Mai, X. Qin, N. O. Silva, J. Molina, and A. A. Cárdenas, "Uncharted networks: A first measurement study of the bulk power system," in *IMC '20: ACM Internet Measurement Conference*, Oct. 27, 2020. DOI: 10.1145/3419394.3423630.

[13] C. Irvene, T. Shekari, D. Formby, and R. Beyah, "If i knew then what i know now: On reevaluating dnp3 security using power substation traffic," in *Proceedings of the Fifth Annual Industrial Control System Security (ICSS) Workshop*, Dec. 2019. DOI: 10.1145/3372318.3372324. (visited on 06/01/2020).

[14] International Electrotechnical Commission (IEC), "Iec 60870-5-101:2003, Telecontrol equipment and systems - part 5-101: Transmission protocols - companion standard for basic telecontrol tasks," International Electrotechnical Commission (IEC), Tech. Rep., Feb. 7, 2003. [Online]. Available: https://webstore.iec.ch/publication/3743 (visited on 03/03/2022).

[15] ——, "Iec 60870-5-104:2006, Telecontrol equipment and systems - part 5-104: Transmission protocols - network access for iec 60870-5-101 using standard transport profiles," International Electrotechnical Commission (IEC), Tech. Rep., Jun. 13, 2006. [Online]. Available: https://webstore.iec.ch/publication/3746 (visited on 03/03/2022).

[16] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for scada networks," in *Proceedings of the SCADA security scientific symposium*, vol. 46, Citeseer, 2007, pp. 1–12.

[17] N. Goldenberg and A. Wool, "Accurate modeling of modbus/tcp for intrusion detection in scada systems," *international journal of critical infrastructure protection*, vol. 6, no. 2, pp. 63–75, 2013. DOI: 10.1016/j.ijcip.2013.05.001.

[18] J. Verba and M. Milvich, "Idaho national laboratory supervisory control and data acquisition intrusion detection system (scada ids)," in *2008 IEEE Conference on Technologies for Homeland Security*, 2008, pp. 469–473. DOI: 10.1109/THS.2008.4534498.

[19] B. Zhu and S. Sastry, "Scada-specific intrusion detection/prevention systems: A survey and taxonomy," in *Proceedings of the 1st workshop on secure control systems*, vol. 11, 2010, p. 7.

[20] B. D. Gregg, *Chaosreader*, version 0.96, Jun. 15, 2014. [Online]. Available: https://github.com/brendangregg/Chaosreader.

[21] D. Fauri, B. de Wijs, J. den Hartog, E. Costante, E. Zambon, and S. Etalle, "Encryption in ics networks: A blessing or a curse?" In *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2017. DOI: 10.1109/SmartGridComm.2017.8340732.