Exposed Infrastructures: Discovery, Attacks and Remediation of Insecure ICS Remote Management Devices

Sasaki, Takayuki; Fujita, Akira; Hernandez Ganan, C.; van Eeten, M.J.G.; Yoshioka, Katsunari; Matsumoto, Tsutomu

DOI
10.1109/SP46214.2022.9833730

Publication date
2022

Document Version
Final published version

Published in
Proceedings - 43rd IEEE Symposium on Security and Privacy, SP 2022

Exposed Infrastructures: Discovery, Attacks and Remediation of Insecure ICS Remote Management Devices

Takayuki Sasaki*, Akira Fujita*†, Carlos H. Gañán‡*,
Michel van Eeten‡*, Katsunari Yoshioka*, and Tsutomu Matsumoto*
*Yokohama National University, †National Institute of Information and Communications Technology, ‡TU Delft

2022 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-1316-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SP46214.2022.9833730

***Abstract*—Geographically distributed infrastructures, such as buildings, dams, and solar power plants, are commonly maintained via Internet-connected remote management devices. Previous studies on detecting and securing industrial control systems (ICS) have overlooked these remote management devices, as they do not expose ICS-specific services like Modbus and BACnet and thus do not show up in Internet-wide scans for such services. In this paper, we implement and validate a discovery method for these devices via their Web User Interface (WebUI) and detect 890 devices in Japan alone. We also show that many of these devices are highly insecure. Many allow access to the status or even the control over industrial systems without proper authentication. Taking a closer look at three prevalent remote management devices, we discovered 13 0-day vulnerabilities, several of which were rated as medium or high severity. They have been responsibly disclosed to the manufacturers. By using honeypots that imitate these systems, we show that over time, only a small number of attackers enter these systems, but some do change critical parameters. Attackers appear to interact more with the system when more facility information is displayed on the WebUI. Finally, we notified operators of 317 vulnerable remote management devices by email and telephone. We reached 212 persons in charge of the devices and received confirmation that our method had correctly identified the device. 50% of the persons in charge of the devices stated that they mitigated or will mitigate the problem. We confirmed their actions via a follow-up scan for vulnerable devices and found that measures were taken for 58% of the devices when we could reach the persons in charge of the device.**

I. Introduction

Over the past decade, researchers have uncovered tens of thousands of industrial control systems (ICS) connected to the Internet [1]. This is widely regarded as a bad security practice, also by the manufacturers [2]. The devices control sometimes critical industrial equipment, yet their protocols were never designed for security in the hostile environment of the open Internet [3]. To make matters worse, many of them remain unpatched for years [2], [4]. The overarching goal of research on discovering Internet-connected ICS is typically to raise awareness, inform system administrators and encourage them to remediate the risks associated with the exposed devices.

Prior research has focused on finding hosts via exposed ICS-specific services like Modbus [5], Siemens S7 [6] and BACnet [7]. This approach means one class of ICS is typically overlooked: remote management devices. Such devices are used in geographically distributed infrastructures, such as buildings, dams, and solar power plants. The devices are positioned in front of the SCADA systems, shielding those systems from the Internet yet allowing operators to remotely monitor and manage them, typically via a web interface. In many cases, they are connected to the Internet via a built-in or an external mobile network access point. These devices will not show up via current methods for discovering ICS, since they do not run Internet-facing ICS-specific services. In the absence of reliable discovery methods, we know little about this population of remote management devices. It is unclear what their prevalence is and what security threat they pose. Their function is to be Internet-connected, so their mere presence on the Internet does not necessarily imply a bad security practice. That being said, if they can be compromised, then attackers would be able to monitor and potentially manipulate ICS devices.

In this paper, we provide an in-depth look at the risk posed by this poorly understood class of ICS by empirically analyzing four aspects of these devices: prevalence, vulnerability, attacker behavior and remediation. For measuring prevalence, we present a novel method to detect remote management devices via semi-automatically generated fingerprints for their web interfaces from scan data. We conduct our initial scans on domestic networks and discover a population of 890 devices, which includes devices used in electric power plants and water gates. We confirm that this approach discovers a previously uncovered class of ICS by comparing our findings to Shodan, a leading industry sources. Over 98% of these discovered devices are not identified as ICS in these sources.

We then take a closer look at the security of the discovered devices. We find that 61% of the devices have no authentication and 24% of the devices are running with unpatched vulnerabilities. We purchased three of these devices and ran standard security tests on them. We discovered 13 0-day vulnerabilities which received CVSS v3 base scores ranging from 3.5 (low) to 9.3 (high). The new vulnerabilities were disclosed to the manufacturers, all of whom have acknowledged the vulnerabilities, requested CVE identifiers and developed patched firmware. Next, we explore if and when such devices are found by attackers by analyzing long-term honeypot data. While Shodan, Censys and Zoomeye discovered our devices within a month, attacker traffic did not increase until much later, when one device was listed on a hacker forum. A fraction of the visitors used the contact details listed on the web

DOI 10.1109/SP46214.2022.00076

interface to notify us about the vulnerable devices. We find that honeypots that display more (fictitious) information about who owns them and where they are located, receive longer visits and more interactions from attackers.

Finally, in collaboration with a government organization we ran a notification campaign to warn the device operators for which we were able to identify the contact points. Prior notification studies—e.g., [8], [9], [10], [11]—sent their notifications to network operators via the listed abuse contact in IP WHOIS records. In contrast, we conducted phone calls and present the first study to directly contact the organization operating the device as well as measure remediation rates after notification. We were able to identify the device operators in 66% of the cases and tried to contact them via phone. Where we were able to speak with the operators, they confirmed the presence of the detected device. This ground truth increases the confidence in our discovery method. About 50% of the operators stated that they would remediate the situation. Follow-up measurements are complicated because of dynamic IP address allocation, but we were able to confirm the improvements in 58% of the cases where we could have reached the persons in charge of the devices. All in all, the effectiveness of the notifications is in line with prior work on high-profile campaigns like website hijacking [12].

In sum, we make the following contributions:

- We implement and validate a novel detection method for ICS devices based on their web interfaces. This discovers a population of devices overlooked in leading industry sources and prior research on ICS.
- We observe that at least 61% of the detected devices are insecure. Testing three specific devices, we discover 13 zero-day vulnerabilities, which we responsibly disclosed to the manufacturers.
- We observed only small attack volumes against these devices over the course of 30 months, even though they were indexed by Shodan and Censys. Some attackers were willing to change critical settings.
- We present the first study directly engaging vulnerable ICS operators and find that notifications lead to high remediation rates.

The remainder of this paper is structured as follows. In Section 2, we briefly explain the functionalities of remote management devices and use cases, then in Section 3, we propose a method to scan the devices and show 890 remote management devices are accessible from the Internet. Then, we perform penetration tests against 3 devices and identify 13 zero-day vulnerabilities in Section 4. Next, we observe the attacks against the devices using a honeypot in Section 5. To remediate the vulnerable devices, we conduct notification activities to the device operators in Section 6. We discuss ethics and related work in Sections 7 and 8, respectively. Finally, we conclude the paper in Section 9.

II. Remote Management Devices

ICS consists of devices with control and monitoring functions, for example controlling the temperature in a chemical plant using a PLC (Programmable Logic Controller) and DCS (Distributed Control System). To monitor the status of the overall system, SCADA (Supervisory Control And Data Acquisition) are used.

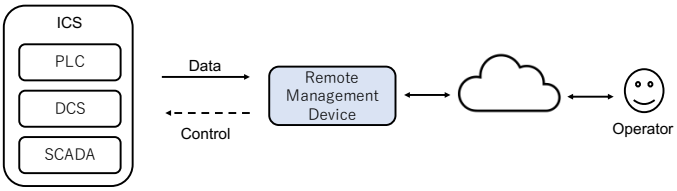

Fig. 1. Infrastructure and a remote monitoring device

Remote management devices are positioned between ICS devices and the Internet (Figure 1). Typically, they are used in situations where ICS is geographically dispersed and needs to be operated remotely. They can be found in critical infrastructures, like the energy and dams sector [1], but also in other facilities like office buildings and shopping centers. An example of a remote management device is NEC's Calsos CSDJ/CSDX Series [13]. It has analog and digital input channels to record data on ICS equipment operation status, failure status, water levels and flow rates. It has analog and digital output channels to control the ICS devices.

Remote management devices typically have two types of interfaces using IP protocols:

- Internet-facing interface for human operators. The remote management devices provide a GUI that can be accessed via a Web browser. The GUI typically includes tables and graphs that illustrate the state of the monitored facility and in some cases even allow remote operators to change the digital/analog outputs of the devices. Some devices can also be configured via the interface, e.g., changing the name of the device, the name of the monitored facility, email address for receiving emergency alerts, etc. In some cases, they also provide command line interface (CLI) such as Telnet or SSH for detailed configuration.
- ICS-facing protocol interfaces for data collection. Some remote management systems support ICS protocols such as Modbus to collect information in a machine-readable manner. The ICS protocols are only meant for communicating with the internal control system behind the management device and are not usually exposed to the Internet, except when misconfigured.

The remote management device is usually connected with the Internet via a mobile network because the devices are geographically distributed and hard to be physically accessed. They are often connected via a mobile router with a SIM card to connect to the mobile network. Some devices internally contain a SIM card to directly connect to the mobile network.

Given the way these devices are designed and deployed, we can identify the security risks that they potentially pose if

[1]National Institute of Standards and Technology (NIST) defines 16 critical infrastructure sectors. See: https://www.nist.gov/cyberframework/critical-infrastructure-resources

they are not adequately secured. In terms of confidentiality, the device might contain sensitive operational information. As for integrity, the correct operation of the associated infrastructure could be disturbed if attackers can modify the configurations or the data sent from the devices to the operators. Finally, in terms of availability, the attacker could perform a DoS/DDoS attack against the device or otherwise disrupt its availability, thus interrupting the monitoring and control of the infrastructure. In case of large disasters such as typhoons, heavy rain, earthquake, and volcanic eruption, the monitoring capability is essential to mitigate the damage of the disasters and to maintain the operation of the infrastructure.

III. Device discovery

To understand the risk associated by remote management devices, we first need to develop a method to discover them. We propose and test a new detection method.

A. Discovery method

Given that remote management devices do not expose ICS protocols to the Internet, unless configured incorrectly, they will not be detected in the common Internet-wide scan methods, as those methods rely on protocols such as ModBus and BACnet [1]. On the other hand, these devices do have a web user interface that identifies them as part of an ICS network. The challenge is how to distinguish these devices among the massive number of hosts with web interfaces.

Our detection method is based on an iterative process that is visualized in Figure 2. Since Internet-wide scans would identify an overwhelming number of WebUIs, we start our process by selecting specific networks where the presence of ICS is more probable, so we can create a seed of WebUI's signatures. Remote management devices are not going to be uniformly distributed across all networks. In light of their functionality, we expect a higher concentration in mobile data communication networks. We select such networks as the starting point to create the initial seed of signatures.

We scan the selected networks and collect the WebUIs present there. The detection approach is based on the intuition that remote management devices will share highly similar WebUIs, while regular websites and the like have higher entropy due to the heterogeneity of the information they contain. There will also be other Internet-enabled appliances, such as general IoT devices, digital video recorders and IP cameras, that will also form clusters with highly similar WebUIs. To differentiate remote management system from these Internet-enabled appliances, we leverage the fact that remote management devices often have a customized field in their WebUI. For example, there is typically a field in their WebUI where the owner or the operator of the device can input the name and location of the monitored facilities as shown in Figure 14 (a). In other words, we expect to find the remote management devices between clusters with less heterogeneity (e.g., IoT devices, default pages of HTTP servers and error pages such as '404 Not Found') and clusters with more heterogeneity (e.g., regular websites).

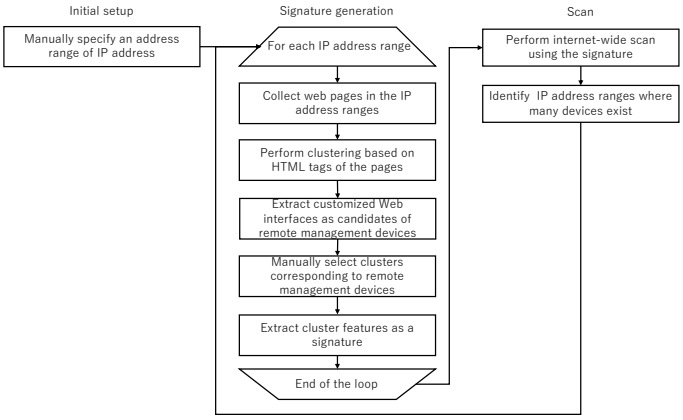

Fig. 2. Web-based scan method for remote management devices

We manually verify that the selected clusters indeed belong to remote management devices and then extract features of each cluster to be used as a signature for that device type. This signature can then be deployed in Internet-wide scans. If those scans result in networks with a high density of devices, then that network might host other remote management devices as well. That network could then be selected for the next iteration of the signature generation process.

B. Implementation

We implement our approach by selecting 4 ISPs providing mobile networks as the starting point for the signature generation. In the later rounds, we scan both mobile and wired ISPs. Using `Nmap`, we scan the IPv4 address ranges of these ISPs on Web-related ports (TCP port 80, 81, 443, 8000, 8080, 8888, and 4443). The IP address ranges are obtained from publicly available AS number and IP address assignment information. If a WebUI is detected, we grab the HTML source using Selenium [14]. Here, we discard HTTP responses with HTTP error code from 400 to 599. Note that we use a full browser to collect the web content as scan tools like `Nmap` and `Zmap` do not handle JavaScript and Web redirection in the same manner as a browser, which could lead to relevant Web content be missed.

Next, we extract and concatenate HTML tags from the obtained HTML source code to represent the structure of the WebUI and perform single-linkage clustering on them based on similarity calculated by `ssdeep`, a fuzzy hash algorithm [15]. Since remote management devices of the same or similar type are expected to have similar WebUI structures, they tend to form a cluster. Specifically, we perform clustering as follows: (i) to generate input of cluster algorithm, we convert HTML code to lower cases and extract HTML tags that represent structure of the WebUI; (ii) we generate a dendrogram based on a single-linkage method and euclidean distance. Here, we define a distance by subtracting the `ssdeep` similarity, whose range is from 0 to 100; and, (iii) we cut the dendrogram at height 30 to obtain clusters.

To perform the cut operations, we use `cut_tree` functions provided by `scipy.cluster.hierarchy` package [16].

Customized words are unique words that only appears once in a cluster. To extract customized words, we: (i) remove HTML tags from HTML codes and obtain contents; (ii) extract (Japanese) strings that are expected to be used for facility names. To extract strings like facility names, we leverage MeCab (Yet Another Part-of-Speech and Morphological Analyzer) [17] and extract strings comprising nouns, prefixes, or symbols. (iii) remove words including time and date, which are unique but not customized words. For the same reason, we also exclude the IP address of the device; and, (iv) from the words of all HTML codes in a cluster, we select words each of which appears only once as customized words. We performed the above process on all clusters and selected clusters each of which has at least one device with a unique word expected as a facility name. Clusters without unique words are discarded as general IoT devices. Finally, we manually check the contents of the WebUI of each cluster to decide if they are indeed remote management devices.

After finding a cluster of remote management devices, we generate a device signature by manually extracting a unique string that only appears in the WebUI of the devices in the cluster. Note that we generate signatures based on content that can be obtained by zmap/zgrab, so that the signatures can be used to efficiently match with Censys scan results, rather than having to run full active scans with Selenium. The list of devices is shown in Table XII in the Appendix.

To scale up the detection of devices using the seed of signatures from the initial set of four mobile ISP networks, we leverage existing datasets of scan engines such as Censys. We search for the signatures within Censys's scan results. For each match, we visit the corresponding IP address with a full browser to obtain the complete WebUI content. From these, we extract the unique strings from the customized fields for all detected devices, as they contain valuable information regarding the owner, operator, and location of the device.

The final step is to identify "hotspots" of IP address ranges where devices are concentrated. These networks could then be used as the input for the next iteration of the detection approach. The threshold for selecting hotspots has to take into account the size of network and the type of device. Using these parameters, we selected three hotspot networks and conducted another cycle of signature generation for those networks.

We continuously conducted the iterations of our Web-based scan process on Japanese IP address ranges from July 2019 to November 2020. Figure 3 shows how the iteration of signature generation and Internet-wide scan could increase the number of identified devices and their models. In total, we performed 39 rounds and generated 23 signatures. At the final rounds, we performed Internet-wide scans using the signatures leading to identifying 890 remote management devices.

During the above rounds, we generated 13,674 clusters from HTML codes in hotspot networks. We selected clusters where there is at least one device with a customized word and got 408 clusters. By manual checks, we identified that 135 clusters

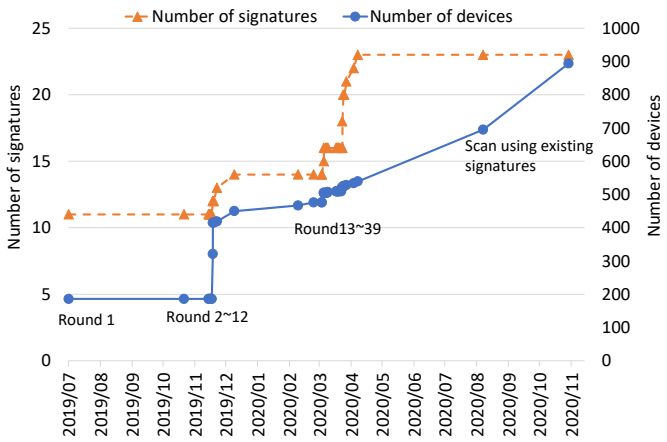

Fig. 3. Round of signature generation and scan

(33%) were remote management devices. By removing duplicates of the same device models in 135 clusters, we identified 23 device models and got their signatures. 273 clusters were false positives each of which has a unique word(s), but the words were neither facility names nor locations of the devices. The false positives were caused by the following words that are misrecognized as customized words:

- Minor changes caused by firmware version differences.
- Customization of WebUI by a manufacturer to distinguish these devices from the OEM version.
- VPN login forms that have customized words such as company names.
- Word differences caused by language settings.

C. Results

During a period of 17 months, we conducted scans with 39 rounds which led to the generation of signatures for 23 device models. Among them, we identified the model names and manufacturers of 21 devices. As for the remaining two devices, we identified that the devices were used for monitoring infrastructure, but we were not able to identify the corresponding model. The detailed device list is shown in Table XII in Appendix.

Using the generated signatures, we detected 890 devices with customized WebUIs. Customized words of 228 devices did not include the name of the facilities and/or organizations; hence the devices were categorized as "unknown". For example, customized words of these devices showed only the device type such as "alert system" or a coarse-grained location such as a city name. Then we manually checked the facility names and identified 473 remote management devices that appeared to be used for critical infrastructure (Table I). (In the later notification phase, we confirmed these findings were correct, see Section VI.) An astonishing number of power plants and water treatment facilities were running an exposed WebUI with the actual name of the facilities. Some of the devices were also used in non-critical facilities such as schools and buildings.

After the remote management devices were discovered, we did not seek information about the systems behind the

TABLE I
FACILITIES WHERE THE REMOTE MANAGEMENT DEVICE ARE USED

	Type of facility	# Devices
Critical	Power plant	223
	Water treatment	224
	Medical center	2
	Transport	7
	Waste treatment facility	11
	Public facility	6
	Subtotal	473
Non-critical	School	78
	Factory	7
	Home buildings	36
	Hotel	6
	Shops	7
	Others	55
	Subtotal	189
Unknown		228
Total		890

device because of ethical reasons. We did not want to engage with actual ICS devices and risk potential disruptions. However, during the notifications and interviews (Section VI), we confirmed that at least a portion of the remote management devices were in fact connected to ICS, such as water gates. Specifically, we visited five facilities (three water treatment facilities, a hospital, and a solar power plant) for field surveys. In all five cases, we confirmed that the remote management devices are connected to the infrastructure.

D. Comparison against industry sources

We compare our results to those of a leading industry source that identifies ICS. Figure 4 shows the overlap of the devices deployed in critical infrastructures in Japan found by our method compared to devices with open ICS ports identified by Shodan[2]. The results of our method were obtained on November 15th 2020, while Shodan uses historical data.

From Figure 4, it is clear that our method can identify many devices deployed in critical infrastructures that are not classified as ICS by Shodan. Specifically, 870 devices that we identified as remote management devices are also found by Shodan, but these were not tagged as ICS because they had no open ICS ports, except for 14 devices. We found 17 devices also ran internet-facing ICS protocols, 14 of which were labeled by Shodan as ICS. In sum, our method effectively identified ICS remote management devices in critical infrastructures that are overlooked by the leading industry source.

E. Global device discovery

To evaluate the capability of global device discovery, we conducted a one-round scan[3] towards IP address ranges that were not included in the signature generation phase. We

[2]Shodan labels the device with the following protocols as ICS, thus we also checked the same ports. Modbus, S7, DNP, Fox, BACnet, EtherNet/IP, GE-SRTP, HART, PCWorx, MELSEC, FINS, Crimson v3.0, CODESYS, EC 60870, ProConOS. https://www.shodan.io/explore/category/industrial-control-systems

[3]Here, we removed a Japanese-specific processing using MeCab.

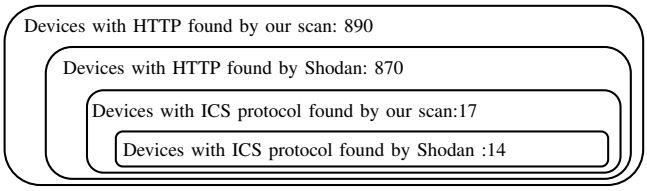

Fig. 4. Relationship between devices identified by our method, those open ports identified by Shodan

selected a mobile ISP in the US and got 37 clusters. Within these clusters, we identified 8 new device models. Using the eight signatures, we found 3,875 devices in total. These devices were deployed at infrastructures such as wind power stations and airports. The list of devices and frequency are shown in Table XIII in the Appendix. There is no overlap between devices discovered by the domestic scan and those by the global scan. It means that the scan method is not over-fitted to discovering Japanese devices. We discuss generalizability of the scan method in Appendix-B. We plan to notify via CERT of each country, rather than conduct phone calls, due to the difficulty and cost of identifying and contacting foreign operators. We have notified the US-CERT as a first step, since more than half of the devices are in the US. We were informed that US-CERT notified the owners of the discovered devices about the security issue.

F. Summary of device discovery

We have developed a new web-based scan method and identified 23 device models and 890 remote ICS management devices in domestic networks. In addition, we identified eight additional device models and 3,875 devices in a global scan. There is little overlap of identified devices between our Web-based scan and ICS protocol-based scan by Shodan.

IV. DEVICE SECURITY

Contrary to many other types of ICS, remote management devices are meant to be Internet-connected. In that sense, exposure does not immediately imply a threat. To understand the risks of the exposed remote management devices, we explore their security posture from three aspects: insecure configurations, unpatched vulnerabilities and zero-day vulnerabilities.

A. Insecure configurations

A fundamental type of insecure configuration is the absence of effective access controls. WebUIs of remote management devices typically provide web-form authentication or HTTP basic authentication. We found that 539 (61%) out of the 890 devices we discovered (Section III) allowed accessing their WebUI without any authentication. This percentage of missing access controls is twice higher than the one reported by a recent study on ICS with OPC UA [3].

WebUIs often provide the internal status of the facilities monitored by the connected sensors—for example, the water level or gas concentration level. They also provide functionality to configure the monitoring settings, including who should

be the recipient of emergency email alerts. In the worst case, unauthenticated devices even provide control options of the facilities, such as opening and closing a water gate, starting a water pump, or turning off the power to the facility.

Even when the WebUI is protected by an authentication mechanism, this could consist of factory-default passwords or weak passwords. For ethical reasons, we did not test which fraction of the 39% of devices with authentication enabled was suffering from this problem, but we did conduct interviews with the operators of these devices during the notification stage of our study (Section VI). During those interviews, 34% of operators admitted that they used either a default or an easy-to-guess password.

Another configuration issue relates to whether the interface displays information about the facility or the operating organization. Of the 890 discovered devices, 712 (81%) exhibited the name of the facilities or related organizations. This information would provide attackers with an idea about what kind of facilities these devices are managing and thus the value of this target. As we will show in Section V, the presence of facility names in the WebUI of our ICS honeypots does attract more engagement from visitors.

B. Unpatched vulnerabilities

Device manufacturers do not have direct channels to the operators because they commonly provide their products via sales companies or system integrators. In addition, the typical remote management devices have neither automatic updates nor a function to inform about firmware updates. This means they cannot notify operators about recommended updates. From our interviews we also learned that none of the discovered devices support automatic updates—which in itself is understandable, given the critical function of ICS devices.

To explore the degree in which known vulnerabilities are patched, we selected a device model from a major vendor where we could distinguish between patched and vulnerable firmware versions from the WebUI. We select CSDJ of NEC Calsos since its old firmware has known vulnerabilities, and it is the fourth largest number of devices identified by our scan.

The older firmware for models CSDJ contains two known vulnerabilities [18] discovered in 2018. These are patched in the current firmware. Of the 71 CSDJ devices that we discovered, 17 (24%) were running on the vulnerable firmware. All in all, we found that the majority of the discovered devices suffered from serious security flaws.

C. Zero-day vulnerabilities

In light of the concerns around poor security in remote management devices, we also assessed the risk of unknown vulnerabilities—i.e., zero-day vulnerabilities. We conducted penetration tests against the WebUIs of three major devices discovered in Sect. III. Specifically, we tested SolarView compact that had the largest number of devices identified by our scans. We also tested Calsos CSDJ that was the third largest, and the DL8, the sixth most frequent device.

The penetration test consisted of Nmap [19] for service discovery and OpenVAS [20] for vulnerability scans. We then manually investigated the HTML and JavaScript code of the Web interface and tested them for prevalent vulnerabilities such as cross-site scripting and improper access control. We also investigated the version numbers of their components, such as embedded servers, to check whether these were running a vulnerable version. Specifically, we investigated HTTP server name and version number from the HTTP header. When a device supported other protocols, such as FTP, we tested these as well.

For just these three devices, we identified 13 zero-day vulnerabilities, 12 of which have been published as CVE entries at the time of the paper submission. They range in severity from 3.5 to 6.3. Specifically, SolarView had vulnerabilities of a directory listing, improper access control, and an OS command injection. In addition, we found usage of vulnerable HTTP and FTP server versions. DL8 had a privilege escalation vulnerability that allows attackers to control digital and analog outputs, which means that ICS devices behind the remote management device can be manipulated. Calsos CSDJ had improper access control for reports of the device status. The details of the vulnerabilities are shown in Table XI in the Appendix.

We followed responsible disclosure practices and notified the vulnerabilities to the device manufacturers 4 months before the submission of this paper. We also notified the Information and Communication Technology - Information Sharing and Analysis Center (ICT-ISAC) and the national CERT. For all three devices, new firmware has been released that includes patches for the reported vulnerabilities. The release of the new firmware has been announced on official web pages of the manufacturers as well as on those of the national CERT. We also conducted conversations with the device manufacturers of the above three devices. During the conversations, the manufacturers promised that they planned to notify their wholesales partners, via which their products are sold to users. All three manufacturers confirmed that they do not have direct sales channels to end users and thus the effectiveness of security notifications could be limited.

D. Summary of device security

We found that 539 (61%) out of the 890 remote management devices allowed accessing their WebUI without any authentication. Furthermore, of the 890 discovered devices, 712 (81%) exhibited the name of the facilities or related organizations. In addition, we performed penetration tests and identified 13 zero-day vulnerabilities for three devices, some of which are critical for the operation of the infrastructure. We informed the manufacturers of these vulnerabilities. The manufacturers fixed the vulnerabilities and released new firmware.

V. Attack observations

To investigate actual attacks against insecure remote management devices, we develop a honeypot that mimics the typical misconfigured setup of remote management system.

A. Honeypot architecture

We show the architecture of our honeypot in Fig 5. The honeypot is meant to mimic insecure facilities that are remotely monitored and managed by remote management devices. There are three types of lure services exposed to the Internet: WebUI, Telnet, and ICS services. Note that such an insecure setup of the devices is not fictional but resembles the services we found during the Internet-wide scans. The details are described in Appendix A.

In the experiment, 30 static IP addresses in a single AS were assigned to the honeypots and each address imitates a different facility. The purpose of this honeypot experiment is to assess what kind of threats these remote management devices are facing especially when they are insecurely operated. In order to efficiently observe possible threats, we particularly focus on the following points:

Attracting attackers: We use a real Programmable Logic Controller (PLC) with a dummy ladder program running as well as a real remote management device. We expose the names and types of fictitious facilities in the WebUI of the management devices as if they were managing important facilities owned by telecom carriers, banks, airports, railway companies, power plants, gas suppliers, government, hospitals, water companies, transport companies, chemical plants, credit card companies, oil companies, schools and shopping malls. Moreover, as we will show later, we confirm that these devices are included in the database of Shodan, Censys, and Zoomeye that could be a channel for possible attackers to find them.

Filtering automated accesses: We are not interested in the large number of automated attacks and research scans against the honeypot. We filter them out and selectively monitor behaviors of non-automated attacks. For WebUIs, we set criteria to distinguish human accesses using a full browser. As for ICS services, we made a signature of the existing ICS discovery tool widely used for Internet-wide scan for excluding them and focused on unique accesses that are compliant with the provided ICS service, which implies that the accessor uses a genuine engineering tool to configure the PLC. As for Telnet, we intentionally expose its login credential in the WebUI so that only those who have accessed the management page or have obtained the credentials from somewhere else could login to the honeypot for further action. Note that such improper exposure of credential does happen in practice as we confirmed in our investigation.

Profiling visitors: Accesses using CLI like Telnet could provide richer attacker profiles, such as their technical skills, terminal settings, and decision making behaviors. For WebUI, we deployed a tracking mechanism using JavaScript-based browser fingerprinting and a cookie to associate longitudinal accesses from the same actors. We also intentionally included Email contact details of the fictitious system operators, hoping some ethical visitors prove their good intention by contacting us. Moreover, we also disclosed a Telnet password to access a Telnet service of the honeypot (see Figure 13 in Appendix).

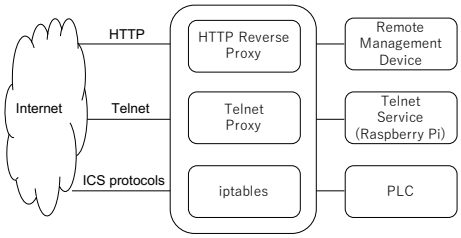

Fig. 5. Honeypot architecture

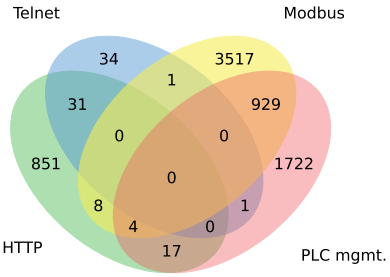

Fig. 6. Overlap of visitors between protocols — The numbers in the Venn diagram denote the number of visitors of each protocol.

B. Results

1) Honeypot observations overview: Table II summarizes our honeypot observations.

- **HTTP.** We ran the WebUI honeypot from August 31st 2018 to January 8th, 2020 and observed over 5 million HTTP requests. We identified 174,244 requests from full-browsers and 822 unique visitors. We observed that attackers actually operated the WebUI, for example, changing device parameters which could potentially affect the functioning of ICS.
- **Telnet.** We operated the Telnet service from March 16th, 2019 to January 8th, 2020 and observed 167 telnet sessions from 67 unique visitors using credentials that we intentionally exposed on the WebUI.
- **Modbus/PLC management protocol.** We ran a Modbus service and a PLC management service from August 31st 2018 to January 8th, 2020. We observed 205,822 TCP sessions for Modbus and 104,158 TCP sessions for the PLC management service.

Figure 6 shows the overlap of visitors between services based on IP addresses. We can identify two types of visitors: HTTP/Telnet visitors and Modbus/PLC management visitors. There are few overlaps between WebUI/Telnet accesses and Modbus/PLC management accesses. In this section, we focus on WebUI/Telnet visitors.

Through the honeypot observations, our key findings are:

- Internet-exposed WebUIs of remote management devices were attacked (Sec. V-B2)
- Honeypot visitors show some level of security expertise (Sec. V-B3)
- Information on the remote management device was exposed at a hacker forum (Sec. V-B4)

TABLE II
HONEYPOT OBSERVATION STATISTICS

Protocol	Period	Statistics			
HTTP 80/TCP	2018/Aug./31 - 2021/Jan./8	SYN packets	HTTP requests	Full-browser acc.	Unique visitors (IP addr. of full-browser acc.)
		5,373,75,746	5,356,630	174,244	822
Telnet 23/TCP	2019/Mar./16 - 2021/Jan./8	SYN packets	Telnet sessions	Login successes	Unique visitors (IP addr. of login successes)
		178,077,667	167	89	67
Modbus 502/TCP	2018/Aug./31 - 2021/Jan./8	SYN packets	TCP sessions		Unique visitors (IP addr. of TCP sessions)
		397,110	205,822		4,459
PLC mgmt.	2018/Aug./31 - 2021/Jan./8	SYN packets	TCP sessions		Unique visitors (IP addr. of TCP sessions)
		252,168	104,158		2,673

- Honeypot visitors changed their behaviors according to what is shown on the WebUI (Sec. V-B5)

2) Observed attacks against the WebUI: We observed that attackers actively interacted with the honeypot WebUI. For example, an attacker tried to change a value of light power and air conditioner status (See Table IX in Appendix for detail). In total, we have observed the following events[4].

- Reset of counter values of digital inputs (230 times by 16 visitors)
- Changes of analog output values (35 times by 15 visitors)
- Changes of ON/OFF statuses of digital outputs (59 times by 14 visitors)

From this result, it is clear that vulnerable remote management devices are critical attack surfaces of infrastructure, and such devices must be mitigated. To remediate the vulnerable devices, we perform a notification campaign (see Section VI).

We also observed a long-term access and multiple accesses to different honeypot IP addresses. For example, an attacker accessed our honeypot four times in two months and the attacker accessed three IP addresses of the honeypot (Table VIII in Appendix A). In total, 66 visitors accessed the honeypot over more than two days, and 131 visitors accessed multiple IP addresses of the honeypot.

3) Observed attacks against the Telnet service: During the observation period, we identified the following typical behaviors among attackers that opened a Telnet sessions.

- Collecting system information by accessing system files/directories and commands such as *apt list*, *netstat*, and *ifconfig* command (84% of the visitors)
- Trying to install applications using *apt install* (11% of the visitors).
- Exploring an internal network using *ping*, *nc*, and *nmap* and so on (13% of the visitors)
- Trying to access an external network using *curl*, *wget* (5% of the visitors)

From the observation, it is expected that the visitors had security knowledge to perform the typical hacking operations.

4) Increase in accesses due to posts at a hacker forum: We observed burst accesses to the honeypot just after two posts about the honeypot at a hacker forum. Each post mentioned a fictional facility name used by the honeypot and an IP address of the honeypot. Table III shows its timeline. Specifically, one of the honeypot's IP addresses, which imitates a remote management device to monitor building facilities of an air control tower, was disclosed in a personal blog of a security researcher on January 9th, 2019. At that time, we did not observe an increase in visitors. On December 8th, 2019, our honeypot was also disclosed at a hacker forum by two posts that were submitted by the same person. We have observed burst accesses to the honeypot after the posts (Figure 7). Moreover a reply to the post said "Open industrial command panel is always cool." From these observations, hackers are expected to be interested in the WebUI of remote management devices. In addition to the burst accesses, we received notifications about the exposed WebUI of our honeypot from ethical visitors. We discuss these notifications in Appendix-A3.

[4]Due to a log collection issue, the period of this analysis is from 31st August 2018 to 7th April 2019.

TABLE III
TIMELINE OF DISCLOSURE BY THE HACKER FORUM AND NOTIFICATIONS FROM HONEYPOT VISITORS

Date	Event
2018/08/31	Honeypot deployment
2019/01/09	A blog article about our honeypot
2019/08/06	Notification from a security researcher
2019/12/08	Two posts at a hacker forum
2019/12/09	Notification from a security company
2019/12/10	Notification from anonymous one
2019/12/17	Notification from CERT

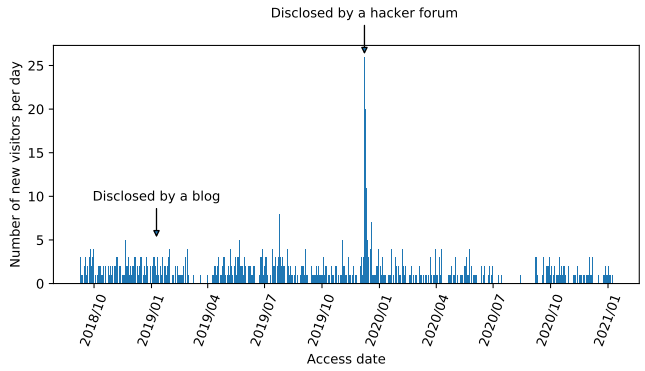

Fig. 7. Number of visitors before and after disclosure by a blog and a hacker forum

5) Differences in visitors' behaviors caused by WebUI contents: We identified that WebUI contents such as a facility name and a picture of a remote management device attract

TABLE IV
DIFFERENCES IN BEHAVIORS OF VISITORS ON EACH TYPE OF WEBUI

Type of WebUI	Daily new visitors per host	Average number of commands	Average visiting duration (Second)
(A)	0.024	5.76	167
(B)	0.027	4.62	120
(C)	0.037	3.46	71
(D)	0.014	1.62*	21

* In case (D), we show the number of login attempts instead of commands.

visitors.

To measure the impact of the WebUI content, we set up four different honeypot WebUIs by changing the contents as follows (See Figure 14 in Appendix for details).

- (A) Full contents. This setup shows a facility name, a device picture, and internal contents such as measured data. Visitors easily identify that a remote management device provides the WebUI.
- (B) Without the facility name. From the device picture on the WebUI, the visitor can identify the WebUI is provided by a remote management device but cannot know what the device manages.
- (C) Without the facility name and the device picture. Careful visitors may identify that the device is a remote management device by its contents such as the structure of the WebUI.
- (D) Login dialog only. This setting provides no information about the Web page.

We deploy (A) from August 31st, 2018 to January 8th, 2021, and (B) (C) (D) from September 10th, 2019 to January 8th, 2021.

Using the above honeypot setups, we analyzed three items: the number of daily new visitors per honeypot, the average number of commands (clicks) on the WebUI, and average visit duration (Table IV). The results show that the more information is provided, the visit duration and the number of commands are increased.

Considering the above results, removing information from WebUIs would avoid attackers by reducing their interest. It is not a perfect security treatment, but to minimize the risk, we recommended removing information such as facility names to the device operators in our notification campaign.

C. Summary of attack observations

We deployed the honeypot of the remote management device and unveiled the attacks against the devices. Analysis of the results showed that visitors were performing long-term access and some tried to change the configuration of the device. In addition, burst access was observed after the disclosure by the post on the hacker forum. We also observed that contents of WebUI such as a facility name and a picture of a remote management device attracted the interest of attackers.

VI. NOTIFICATION CAMPAIGN

To mitigate the risks posed by Internet-exposed vulnerable remote management devices, we performed a notification campaign based on telephone calls.

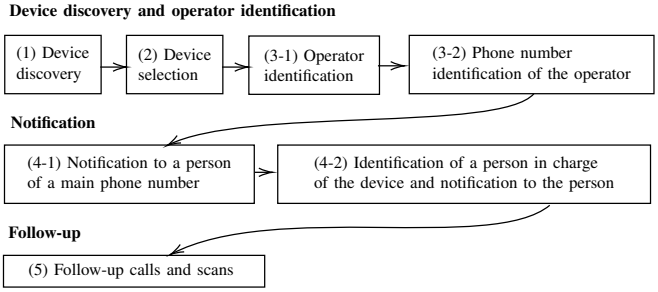

Fig. 8. Steps of the entire notification process

A. Notification procedure

We performed notifications based on manual phone calls and leveraged emails for additional communication, for example, for sending a survey questionnaire and making appointments for telephone calls. To deal with manual operations, we built a dedicated team to handle each case of the notification. The notifications were conducted from October 13th, 2020 to January 29th, 2021. Our notification procedure comprises of 5 steps: (1) device discovery, (2) device selection, (3) identification of device operators and their contact phone numbers, (4) notification, and (5) follow-up (Figure 8).

(1) Device discovery. We performed scans against remote management devices using the method described in Section III.

(2) Device selection. Due to the limited resources of our notification team, we performed triage of the devices. Specifically, we excluded the devices used for small solar power generators and devices used at residences, and we did not notify operators of the excluded devices.

(3) Identification of device operators and their contact phone numbers. We manually identified the device operators using the following information.

- Facility names and/or locations on WebUIs. We searched the names and/or locations using a search engine and identified the operators. If there are multiple candidates with the same facility/location name, we further use the following information.
- Pictures of facilities and/or maps on WebUIs. In the case of devices with pictures and/or maps where the devices were deployed, we also used this information to narrow down the candidates of the operators.
- IP WHOIS information. In addition, we used ISP information described in the network information field of IP WHOIS when it contains location information such as a prefecture name.

Once the operators of the devices were identified, we searched their contact phone numbers based on the operator names.

(4) Notification. We notified the operators of the devices by calling to the identified phone numbers related either to the operators themselves or the organization responsible for the devices. Figure 15 in Appendix shows details of our notification flow. First, we made calls to each organization's

TABLE V
STEPS FROM DEVICE DISCOVERY TO NOTIFICATIONS TO PERSONS IN CHARGE

Notification steps and results	Devices	Operators
(1) Device discovery		
Devices discovered by our scan	890	-
(2) Device selection		
Unselected devices used for low-importance facilities	359	-
Selected devices for notification	531	-
(3-1) Operator identification		
Devices whose operators were unknown	179 (34%)	-
Devices whose operators were identified	352 (66%)	191
(3-2) Identification of phone numbers		
Devices that we were not able to deal with in the notification period due to the limited resources of our notification team	21 (6%)	20 (10%)†
Operators whose contact phone numbers were unknown	14 (4%)	12 (6%)
Operators whose contact phone numbers were identified	317 (90%)	160 (84%)†
(4-1) Notification to a main phone number		
Performed notifications	317	160
(4-2) Notification to a person in charge		
Notifications refused by the operators	5 (2%)	5 (3%)
Cases where a person in charge of the device was unknown or cannot have been contacted	44 (15%)	25 (16%)‡
Cases where the operator did not know the devices	56 (18%)	41 (26%)‡
Notifications performed to the persons in charge of devices	212 (67%)	93 (58%)‡

†‡ There are overlaps since an operator would manage multiple devices

main phone number and explained our notification project and the risks of the Internet-exposed devices. Then, to maximize the efficiency of the notifications, we tried to identify a person that actually operated or managed the device and contact the person. If we could reach the person, we explained the risks of the exposed device and the following mitigations.

- Removing a facility name and/or a location from a WebUI so that the devices do not attract attackers.
- Deploying network access control mechanisms such as a firewall (IP filtering) and a VPN to allow only operators to access the WebUI.
- Changing a password if a password is a default or a weak password.
- Updating firmware if the firmware is old or vulnerable and constructing an organizational framework to keep the firmware up to date.

(5) Follow-up. Finally, we tracked the status of the devices to measure the efficiency of the notifications. Specifically, after the expected date of the mitigation deployment, we made extra calls to ask about the status of the mitigations. In addition, we scanned and tracked the devices using the device signatures and the facility names and checked whether mitigations were deployed or not.

B. Notification results

Table V summarizes our notification result. We discovered 890 remote management devices by our scan method, and then we selected 531 devices. Of the selected devices, we identified operators of 352 devices. Unfortunately, we were not able to identify the operators of 179 devices.

Of the devices whose operators were identified, we identified the contact telephone numbers of 160 operators and performed notifications to them about 317 devices. Unfortunately, we were not able to identify the contact phone numbers of 12 operators. Most operators listened to our notifications. In 5 cases, our notification activities were refused. We explained the situation regarding the exposed remote management devices to persons on the main phone numbers of the organization. Next, we tried to identify and contact the persons in charge who actually managed the device, but in 44 cases, we were not able to reach the persons in charge of the devices. In cases of 56 devices, persons on main phone numbers and persons in charge of the device did not know the remote management devices. One explanation is that we contacted organization that did not actually operate the devices. Though it could also be the case that the person did not know the device due to limited information sharing within their organizations. Finally, we suggested mitigations to 93 operators for 212 devices. During the notifications to the persons in charge, we asked them questions about their responses to our notifications (Table VI).

Removal of the facility name from WebUI. 40% of the operators answered that they performed or planned the proposed mitigation. The major barrier to the deployment of this measure was that the facility names on the WebUI were required for their operations.

Network access control. 25% of the operators answered that they performed or planned to deploy this mitigation. This measure requires a budget to deploy a firewall or a VPN; thus, the budgetary impact is higher compared to other mitigation measures. Note that one operator, who managed three devices, answered that they had already deployed network access control, but actually, the devices were identified by our Internet-wide scan. It seemed that they misconfigured the network access control.

Strong passwords. Approximately one-third of operators had already changed the passwords from the default ones and used strong passwords. The other operators used vulnerable passwords or did not understand the situation of the password setting. 34% of all operators (52% of the operators using vulnerable passwords or not getting a grasp of the setting) answered that they changed or planned to change passwords following our notification. We only got a few negative responses to this recommendation, because password updates do not require an additional budget nor change their operational work flows.

Firmware update. Approximately half of the operators already had used the latest version of the firmware. The other operators used old firmware or did not know the versions

TABLE VI
RECOMMENDED MITIGATIONS AND OPERATORS RESPONSES

Operator's responses	Recommended mitigations							
	Removal of a facility name		NW access control		Strong passwords		Firmware update	
	#Dev	#Opr	#Dev	#Opr	#Dev	#Opr	#Dev	#Opr
Operators deployed the mitigation by our notifications	**46 (22%)**	**21(23%)**	**26 (12%)**	**11(12%)**	**56 (26%)**	**29(31%)**	**14 (7%)**	**10(11%)**
Operators planned to deploy the mitigation	**35 (17%)**	**16(17%)**	**26 (12%)**	**12(13%)**	**6(3%)**	**3(3%)**	**7 (3%)**	**3(3%)**
Operators already deployed the mitigation in advance to our notification	0 (0%)	0(0%)	3(1%)	1(1%)	99 (47%)	31(33%)	122(58%)	42(45%)
Under investigating the availability of the mitigation	67 (32%)	19(20%)	59 (28%)	17(18%)	16(8%)	9(10%)	22(10%)	14(15%)
Unable to deploy the mitigation	31(15%)	16(17%)	52(25%)	26(28%)	5(2%)	3(3%)	19(9%)	8(9%)
No budget	7	3	10	4	2	1	6	3
Current setup is required for the operation	22	12	34	18	1	1	0	0
No person who can handle the issue	2	1	8	4	2	1	6	2
No contract with the system integrator who sets up the devices	0	0	0	0	0	0	7	3
Decide not to deploy the mitigation	10(5%)	9(10%)	22(10%)	14(15%)	8(4%)	7(8%)	8(4%)	7(8%)
Unable to understand importance of the mitigation	0	0	0	0	0	0	1	1
Superior manager's decision	3	2	3	2	3	2	3	2
No actual damage are caused even the measure is not deployed	7	7	19	12	5	5	4	4
No answer in the notification period	23(11%)	14(15%)	24(11%)	15(16%)	22(10%)	13(14%)	20(9%)	11(12%)

of the firmware. 14% of all operators (25% of operators not using the latest version or not getting a grasp of the setting) answered that they updated or planned to update the firmware. The firmware update requires a few technical operations; thus, some operators answered that they were not able to update the firmware because of contractual limitations with the system integrator who set up the device. During the notification, we also asked if the organizations had organizational frameworks in place to keep the firmware latest. 69% of the operators said to have a framework, but 31% of the operator said they did not.

All in all, 47 operators (50% of the operators we notified to person in charge of the devices) managing 101 devices answered that they deployed or planned to deploy at least one mitigation of the above four mitigations.

C. Follow-up scans

During and after the notification period, we conducted follow-up scans against the devices. Note that we did not test the status of the password update for ethical reasons. We also did not check the firmware versions because it is difficult to infer the firmware versions from the information of the WebUI.

The IP addresses used by some devices changed due to DHCP-based IP address assignment, so simply revisiting the IP address would not correctly reflect the actual situation. Therefore, our follow-up scans were performed by Internet-wide scans using device signatures and tracking based on facility names. Devices become undetectable if network access control or removal of facility names are performed. During the follow-up scans, we identified that 29% of the devices changed their IP addresses.

Figure 9 shows the number of detected devices by follow-up scans using the number on the first scan as a reference. We counted devices in three categories: devices without notification (devices used for low important facilities or whose operators were not identified), devices with notifications to the persons in charge, and devices with notifications only to persons on the main phone numbers (notifications did not reach the person in charge). Throughout the follow-up scans, the number of devices without notifications stayed more or less constant and only decreased by 13%. In contrast, our notifications had a significant impact, and devices have been reduced by 58% when we were able to contact the persons in charge of the devices. To test whether the observed differences in remediation are significant between the group with notification and without notification, we conducted a chi-squared test. The p-value of the chi-square test is less than 0.0001 which means that the effect of our notification campaign was highly statistically significant. Even where we were not able to talk to the persons in charge, the presence of the devices was reduced by 23%.

Table VII shows the relation between the responses from the operators and the actual mitigations as confirmed by follow-up scans. We categorize the operators' responses into four types: they answered that they had done the mitigation (*Done* in the figure), they planned the mitigation (*Planned*), they were considering the mitigation (*Under consideration*), and they did not intend to deploy the mitigation (*Won't do)*. 46 operators answered that they deployed at least one mitigation, and actually 98% of them remediated the devices. Surprisingly, 21 organizations that answered that they would deploy neither of two mitigations actually mitigated 22% of the devices.

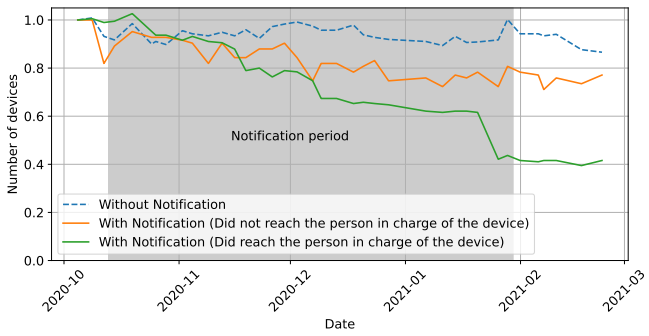

Fig. 9. The number of devices detected by follow-up scans using the first scan as the base

TABLE VII
OPERATORS' RESPONSES ABOUT REMEDIATION AND ACTUAL REMEDIATION RATES

		Responses to removal of facility names			
		Done	Planned	Under Consid.	Won't do
Responses to deployment of NW access control	Done	98% (45 mitigations confirmed by follow-up scans / 46 responses)			
	Planned		51% (18 migrations / 35 responses)		
	Under Consideration			70% (63 mitigations / 90 responses)	
	Won't do				22% (9/41)

D. Comparisons to previous notification studies

Here, we discuss the results of our notification in comparison to other notification projects in terms of reachability and remediation rate.

1) Notification reachability: Where we were able to identify the organization, we were able to get in touch with a large portion of the operators, compared to previous studies that were email-based. Specifically, of the identified operators, we identified 90% of their contact phone numbers and we were able to contact all of the operators using the phone numbers.

In a past notification campaign against DNS zone poisoning [11], at maximum 70.4% of emails sent to the domain owners were undelivered. In another notification campaign against misconfiguration of IPv6 firewall, DDoS amplifier, and exposed ICS services [9], 77% were automated responses and they only got 14% human replies when performing notification based on the email using WHOIS information. In the notification campaign against WordPress vulnerabilities and client-side XSS [10], only 6.4% (1,150 out of 17,916) emails were reached. Compared to these rates, our reachability rate of 90% is significantly higher.

2) Remediation rates: In our notification process, we directly contacted persons in charge of the remote management devices. This process contributed to the mitigation rates, which is 58% out of devices with notifications to persons in charge.

We cannot directly compare the remediation rates with previous studies since the rates depend on the severity of the risks posed by the vulnerability and the complexity and cost of mitigation. A previous project that is somewhat similar to ours, did notifications about open ICS ports [9], and its remediation rate was approximately 18%—i.e., much lower than the rate we observed. Our remediation rate is higher than most previous notification experiments: it was approximately 40% for cross-site scripting and a WordPress vulnerability [10], 33%∼42% for different WordPress vulnerability [21], and less than 20% for DNS zone poisoning [11]. The only campaigns that reported similar remediation rates were on publicly accessible Git repositories (78%∼81%) [21] and Heartbleed (approximately 40%∼90%) [22].

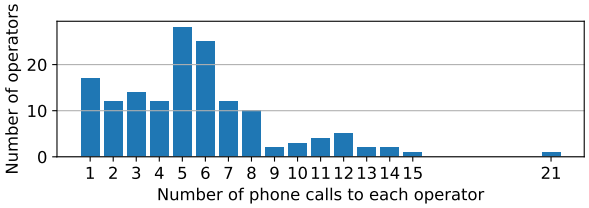

Fig. 10. Number of telephone calls to each operator

E. Cost of telephone-based notification

The notifications using telephone calls require a substantial effort compared to notifications using emails. Even in our notification project, we were not able to deal with 23 devices in the notification period due to the limited resource of our notification team. The following matters increase the cost of a telephone-based notification.

Manual operations. Notification by telephone is hard to automate. To perform notifications, we constructed a dedicated team comprising of four members and we had to manually identify the operators and their phone numbers and make calls. Specifically, the notification was conducted by a full-time supervisor, a full-time phone-call operator, a half-time manager, and a part-time system engineer. The roles were flexibly changed according to the situation of the phone calls.

Multiple phone calls. To perform notification to the right person, we had to call several times. For example, an initial call to the organization's contact point and then a call to the person in charge of the device to present the measures to mitigate the risks. In addition, the number of phone calls increased over time; for example, the person in charge is absent, extra phone calls to ask the status of the mitigations. At maximum, we called 21 times to an operator (Figure 10).

F. Summary of the notification campaign

We performed a notification campaign based on telephone calls. We identified the contact telephone numbers of 160 operators and performed notifications. We reached the persons in charge of the devices and suggested mitigations for 93 operators associated with 212 devices. By follow-up scans, we confirmed that 58% of vulnerable devices were remediated when we were able to reach the persons in charge of the devices. Even when we were not able to contact the persons in charge, the devices were remediated by 23%.

VII. Ethics and responsible disclosure

For responsible disclosure, we have notified the identified zero-day vulnerabilities to the device manufacturers, the national CERT, an industrial consortium, and a governmental entity. After our notifications, the vulnerabilities got fixed and new firmware was released. To preserve privacy, in this paper, we only show statistics and do not disclose information about individual cases. The scans against domestic IP address ranges and the notification campaign were performed under the authorization of the Ministry of Internal Affairs and Communications.

As for ethical considerations of honeypot measurements, our institution does not have an IRB for computer science research, and therefore we could not apply for IRB approval. We designed our study in line with the four principles discussed in Menlo report [23]: Respect for Persons, Beneficence, Justice, and Respect for Law and Public Interest. Particularly, we have carefully considered possible harms that might be done by our honeypot in line with prior studies [27∼36]. Specifically, to avoid using the honeypot for attacks, we blocked outbound traffic from the honeypot to the Internet. With the above measure, we believe we could minimize the harm while gaining substantial benefits by providing insights on malicious activities on the remote management devices by threat actors.

VIII. Related work

In this section, we briefly review the prior work in terms of device discovery, attack observation, and notification.

A. Device discovery

An Internet-wide view of ICS devices has been surveyed by Mirian et al. [1]. They conducted scans against ICS ports using ZMap with customization to perform handshakes of ICS protocols, and they identified from 500,000–800,000 devices for each protocol. Moreover, they have deployed honeypots supporting ICS protocols and observed scan activities conducted by a university and cyber-security companies. A similar survey has been conducted by Xuan Feng et al. [24]. Nawrocki et al. reported that unprotected ICS communications were detected by monitoring traffic at an Internet exchange [25]. As for device discovery using Shodan, Bada and Pete [26] surveyed the cybercrime ecosystem, and they clarified that Shodan was used for finding attack targets.

B. Attack observation

Many types of honeypots to observe attacks against IoT systems and ICS have been proposed. IoTPot [27], [28] is a honeypot that emulates IoT devices. Specifically, IoTpot supports ARM, MIPS, and PPC CPU architectures and exposes a Telnet service. Conpot [29] is an ICS honeypot that supports BACnet and Modbus. A PLC honeypot system emulating Siemens S7-200 [30] has been developed to detect attacks against ICS. Kyle Wilhoit et al. have surveyed attackers against ICS devices using a real PLC (Nano-10) and observed unauthorized accesses [31]. In addition to the above honeypot, many types of ICS honeypots have been proposed [32][33][34][35][36].

The situation of cyberattacks against ICS has been investigated by monitoring devices, botnets, and globally deployed honeypots [2]. Marnerides et al. reported that ICS networks were compromised due to non-ICS devices such as routers, servers, and IoT devices that were used as footholds for lateral movement [37].

C. Notification

Email-based notification campaigns have been conducted in a variety of studies. Cetin et al. identified Mirai-infected devices by combining darknet observations, IoT honeypots, and Shadowserver [38]. To remove Mirai-infected devices, they quarantined the infected devices by collaborating with an ISP and redirected the communication from the devices to a notification web page [8]. Moreover, Cetin et al. have conducted a notification campaign against DNS zone poisoning vulnerabilities to operators of domain servers, domain owners, and network operators [11]. Li et al. have conducted a large-scale notification campaign against three vulnerabilities: misconfiguration of IPv6 firewall, DDoS amplifier, and exposed ICS services such as BACNet and ModBus [9]. They have performed direct notifications by emails on the basis of WHOIS information and indirect notifications via CERT. Ben et al. have conducted a similar notification campaign against WordPress vulnerabilities and XSS vulnerabilities [10].

IX. Conclusion

To make critical infrastructures more secure, we have developed a method for the discovery of remote ICS management devices. We also analyzed their vulnerabilities, observed the attacks using honeypots, and conducted extensive notifications to device operators. For each step, we identify key takeaways.

- **Device discovery.** We have developed a new web-based scan method and identified 890 remote ICS management devices in domestic networks. There is little overlap of identified devices between our Web-based scan and ICS protocol-based scan by Shodan.
- **Vulnerable devices.** Many Internet-facing remote management devices have vulnerabilities in terms of authentication issues or outdated firmware. Moreover, via penetration tests, we identified 13 zero-day vulnerabilities for three devices, some of which are critical for the operation of the infrastructure. We informed the manufactures of these vulnerabilities.
- **Attack observation.** We also observed the accesses to the remote management devices and unveiled that attacks against the devices. Analysis of the results showed that visitors were performing long-term access and some tried to change the configuration of the device. In addition, burst access was observed after the disclosure by a post on the hacker forum.
- **Notification.** We notified operators about the discovered vulnerabilities via phone calls. By follow-up scans, we confirmed that 58% of vulnerable devices were remediated when we were able to reach the persons in charge of the devices.

ACKNOWLEDGMENTS

A part of this research was conducted in "MITIGATE" project among "Research and Development for Expansion of Radio Wave Resources(JPJ000254)", supported by the Ministry of Internal Affairs and Communications (MIC), Japan. A part of this research was conducted in "WarpDrive" project, supported by National Institute of Information and Communications Technology, Japan. The notification project was conducted by a consortium of Yokohama National University, NTT Communications Corporation, and ICT-ISAC Japan with the support from MIC. This work was also partly supported by the Dutch Research Council (NWO) under the RAPID project (Grant No. CS.007) and the "Hestia Research Programme" (Grant No. VidW.1154.19.011).

REFERENCES

[1] A. Mirian, Z. Ma, D. Adrian, M. Tischer, T. Chuenchujit, T. Yardley, R. Berthier, J. Mason, Z. Durumeric, J. A. Halderman, and M. Bailey, "An internet-wide view of ics devices," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, 2016, pp. 96–103.

[2] M. Dodson, A. R. Beresford, and D. R. Thomas, "When will my PLC support Mirai? The security economics of large-scale attacks against ICS," *Symposium on Electronic Crime Research (eCrime)*, 2020.

[3] M. Dahlmanns, J. Lohmöller, I. B. Fink, J. Pennekamp, K. Wehrle, and M. Henze, "Easing the conscience with opc ua: An internet-wide study on insecure deployments," in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC '20, 2020, p. 101–110.

[4] B. Wang, X. Li, L. P. de Aguiar, D. S. Menasche, and Z. Shafiq, "Characterizing and modeling patching practices of industrial control systems," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 1, Jun. 2017.

[5] "Modbus specifications," http://www.modbus.org/specs.php.

[6] "Siemens s7," https://support.industry.siemens.com/cs/document/26483647/what-properties-advantages-and-special-features-does-the-s7-protocol-offer-?

[7] "Bacnet," http://www.bacnet.org/index.html.

[8] O. Çetin, C. Ganán, L. Altena, T. Kasama, D. Inoue, K. Tamiya, Y. Tie, K. Yoshioka, and M. van Eeten, "Cleaning up the internet of evil things: Real-world evidence on isp and consumer efforts to remove mirai." in *NDSS*, 2019.

[9] F. Li, Z. Durumeric, J. Czyz, M. Karami, M. Bailey, D. McCoy, S. Savage, and V. Paxson, "You've got vulnerability: Exploring effective vulnerability notifications," in *25th USENIX Security Symposium (USENIX Security 16)*, Austin, TX, Aug. 2016, pp. 1033–1050.

[10] B. Stock, G. Pellegrino, C. Rossow, M. Johns, and M. Backes, "Hey, you have a problem: On the feasibility of large-scale web vulnerability notification," in *25th USENIX Security Symposium (USENIX Security 16)*, Austin, TX, Aug. 2016, pp. 1015–1032.

[11] O. Cetin, C. Ganan, M. Korczynski, and M. van Eeten, "Make notifications great again: learning how to notify in the age of large-scale vulnerability scanning," in *Workshop on the Economy of Information Security*, 2017.

[12] F. Li, G. Ho, E. Kuan, Y. Niu, L. Ballard, K. Thomas, E. Bursztein, and V. Paxson, "Remedying web hijacking: Notification effectiveness and webmaster comprehension," in *International World Wide Web Conference*, 2016.

[13] NEC Platforms,LTD, "Calsos," https://www.necplatforms.co.jp/en/index.html.

[14] "Selenium," https://www.selenium.dev/.

[15] "ssdeep - fuzzy hashing program," https://ssdeep-project.github.io/ssdeep/.

[16] "Hierarchical clustering (scipy.cluster.hierarchy)," https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html.

[17] "Mecab: Yet another part-of-speech and morphological analyzer (japanese)," https://taku910.github.io/mecab/.

[18] "CVE-2018-0613 and CVE-2018-0614," https://jvndb.jvn.jp/en/contents/2018/JVNDB-2018-000068.html.

[19] "Nmap," https://nmap.org/.

[20] "Openvas (open vulnerability assessment scanner)," https://www.openvas.org/.

[21] B. Stock, G. Pellegrino, F. Li, M. Backes, and C. Rossow, "Didn't you hear me? - towards more successful web vulnerability notifications," in *25th Annual Network and Distributed System Security Symposium (NDSS)*, 2018.

[22] Z. Durumeric, F. Li, J. Kasten, J. Amann, J. Beekman, M. Payer, N. Weaver, D. Adrian, V. Paxson, M. Bailey, and J. A. Halderman, "The matter of heartbleed," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14, 2014, p. 475–488.

[23] "The menlo report: Ethical principles guiding information and communication technology research," https://www.dhs.gov/sites/default/files/publications/CSD-MenloPrinciplesCORE-20120803_1.pdf, 2012.

[24] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun, "Characterizing industrial control system devices on the internet," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, 2016, pp. 1–10.

[25] M. Nawrocki, T. C. Schmidt, and M. Wählisch, "Uncovering vulnerable industrial control systems from the internet core," in *IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1–9.

[26] M. Bada and I. Pete, "An exploration of the cybercrime ecosystem around Shodan," in *The 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS 2020)*, 2020.

[27] Yin Minn Pa Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "Iotpot: Analysing the rise of iot compromises," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. USENIX Association, 2015.

[28] Yin Minn Pa Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "Iotpot: A novel honeypot for revealing current iot threats," *Journal of Information Processing*, vol. 24, no. 3, pp. 522–533, 2016.

[29] "Conpot," https://github.com/mushorg/conpot.

[30] D. I. Buza, F. Juhász, G. Miru, M. Félegyházi, and T. Holczer, "Cryplh: Protecting smart energy systems from targeted attacks with a plc honeypot," in *Smart Grid Security*, J. Cuellar, Ed. Cham: Springer International Publishing, 2014, pp. 181–192.

[31] K. Wilhoit, "Who's really attacking your ics equipment?" *Trend Micro*, vol. 10, 2013.

[32] D. Antonioli, A. Agrawal, and N. O. Tippenhauer, "Towards high-interaction virtual ics honeypots-in-a-box," in *Proceedings of the 2Nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, ser. CPS-SPC '16, 2016, pp. 13–22.

[33] A. V. Serbanescu, S. Obermeier, and D. Yu, "A flexible architecture for industrial control system honeypots," in *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*, vol. 04, July 2015, pp. 16–26.

[34] H. Naruoka, M. Matsuta, W. Machii, T. Aoyama, M. Koike, I. Koshijima, and Y. Hashimoto, "Ics honeypot system (camouflagenet) based on attacker's human factors," *Procedia Manufacturing*, vol. 3, pp. 1074 – 1081, 2015.

[35] A. V. Serbanescu, S. Obermeier, and D.-Y. Yu, "Ics threat analysis using a large-scale honeynet," in *3rd International Symposium for ICS & SCADA Cyber Security Research 2015 (ICS-CSR 2015) 3*, 2015, pp. 20–30.

[36] D. Antonioli and N. O. Tippenhauer, "Minicps: A toolkit for security research on cps networks," in *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, ser. CPS-SPC '15, 2015, pp. 91–100.

[37] A. K. Marnerides, V. Giotsas, and T. Mursch, "Identifying infected energy systems in the wild," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ser. e-Energy '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 263–267.

[38] "Shadowserver foundation," https://www.shadowserver.org/.

[39] "Guerrilla mail," https://www.guerrillamail.com/.

[40] "Unauthorized access to water level monitoring camera and password changed (japanese)," https://scan.netsecurity.ne.jp/article/2018/04/27/40880.html, 2018.

[41] "Unauthorized access to tepco's surveillance cameras (japanese)," https://cybersecurity-jp.com/news/24393, 2018.

[42] "Ongoing sophisticated malware campaign compromising ics (update e)," https://us-cert.cisa.gov/ics/alerts/ICS-ALERT-14-281-01B, 2014.

[43] R. J. Turk, "Cyber incidents involving control systems," Idaho National Laboratory (INL), Tech. Rep., 2005.

[44] "Proof of concept of attack codes (remote code execution of wago e!display 7300t)," https://www.exploit-db.com/exploits/45014, 2018.

[45] K. Wilhoit, "Who's really attacking your ics equipment?" *Trend Micro*, vol. 10, 2013.

APPENDIX

A. Honeypot details

Our honeypot exposed HTTP, Telnet, and ICS services. During the device discovery discussed in Section III, we identified 4 IP addresses with all of an HTTP port, an ICS port, and a Telnet port.

1) Telnet analysis: To profile the attackers, we analyzed the number of Telnet commands and access period. Figure 11 shows the number of Telnet commands per session (Y-axis) and access date (X-axis) of each telnet session. A pair of a shape and color represents one visitor's IP address. Telnet sessions with no input command are omitted in the figure. We can identify two types of access patterns: short-term access and long-term access. As for the short-term accesses, visitors performed burst access to the honeypot on a day. For example, brown gamma(γ), magenta circle (•) and blue diamond (◆), and sky bule x(x) are this type. It is assumed that the visitors intensively surveyed the honeypot on the initial stage and decided that the honeypot is out of their interest. As for the long-term access, one visitor (green triangle (◀)) visited several times on different dates. The numbers of commands per telnet session are approximately 10 and not large compared to the other telnet sessions.

Figure 12 shows examples of commands, from which we can evaluate the skills of visitors. In case 1, the visitor has knowledge about options of *nmap* command and used a "reason" option that outputs a reason for the decisions about service identification. In case 2, the visitor knows a hacking technique about a reverse shell connection using *nc* command and Python. In case 3, the visitor knows the output of *netstat* command includes ESTABLISHED or LISTEN.

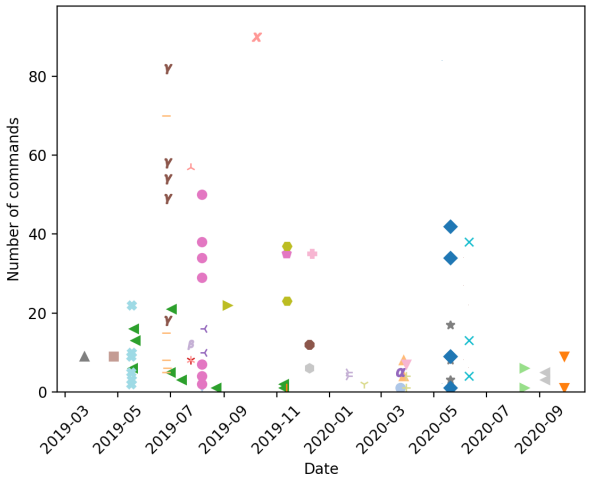

Fig. 11. Number of telnet commands in a session and access date. Each color-shape pair represents an IP address of a visitor. Visitors (γ•◆x) performed burst accessed on a day. A visitor (◀) performed a long-term access.

```
Case 1: nmap -p 1-1023 –reason [An internal IP address]
Case 2: nc [A grobal IP address] 7414 -e /bin/bash
 python -c import socket,subprocess,os;s=socket.socket[snip]
Case 3: netstat -na |grep ESTA
 netstat -na |grep LIST
```

Fig. 12. Example of commands in Telnet service

TABLE VIII
LONG-TERM MULTIPLE ACCESSES

Date	Behavior
September 28th, 2018	An attacker accessed three honeypots (A, B, C). As for honeypot A and B, the attacker visited several pages, and for honeypot C attacker only accessed the top page.
October 1st, 2018	The attacker accessed three honeypots (A, B, C). As for honeypot A, the attacker accessed several pages, and for honeypot B and C, the attacker only accesses the top page.
October 24th, 2018	The attacker accessed honeypot A and C.
November 23rd, 2018	The attacker accessed the top page of honeypot A. After 15 minutes, the attacker visited the honeypot A again and accessed several pages.

2) Analysis of accesses to PLC: In addition to access to HTTP and Telnet services, we have observed access to PLC-specific services.

As described in Section V-A, the PLC deployed in our honeypot is exposed to the Internet and provides services to manage the PLC. When the PLC receives a specific message to the TCP ports of the services, the PLC returns a response including its device information. Using this feature, PLCs can be identified using an Internet-wide scan. A discovery tool that leverages this feature is available on the Internet. Our honeypot observed the scan activities using this tool. We also observed traffic generated by an engineering tool dedicated to the PLC for programming and configuration software.

The attackers using the above tools clearly target the industrial systems with the PLCs, but as discussed in Section V the attackers are not overlapped with the attackers against WebUIs.

3) E-mail notifications from ethical visitors: We received four notifications about the exposed WebUIs of our honeypot. From the notifications, we concluded that ethical security experts accessed the honeypot.

We also disclosed an email address as the contact information of a fictional system operator on the WebUI. As a result, we have received three notifications to the disclosed email address. We received a notification via a domestic CERT.

- Notification 1 was sent to the disclosed email address on August 6th, 2019 from a self-identified sender. This notification mentioned the IP address of the honeypot and the device model name of the remote management device. We confirmed the identity of the email sender and identified him as a hardware security researcher.
- Notification 2 was sent to the email address on December 9th, 2019 by an employee of a cyber-security company. This email mentioned that the IP address was owned

TABLE IX
ATTACK EXAMPLE (EXCERPT OF ACTIVITIES OF A VISITOR)

Time	Behavior
0 sec	Access a top page of a device
(SNIP)	
10 min 21 sec	Access a configuration page of the device
10 min 56 sec	Change a value (Power of light:98.000 → 20.000)
12 min 16 sec	Change a value (Power of light:100.000 → 50.000)
13 min 25 sec	Change air conditioner status:OFF → ON
25 min 55 sec	Access the configuration page of a device
43 min 21 sec	Change a value (Power of light:98.000 → 95.000)
2 hour 8 min 35 sec	Access a log page of the device.
2 hour 13 min 52 sec	Access an event log page.

Fig. 13. Disclosure of telnet credentials on honeypot WebUI

by our university and wondered about the relationship between our university and an air control tower that our honeypot imitates.

- Notification 3 was sent to the email address on December 10th, 2019 from an anonymous person. The anonymous e-mail service (Guerrilla Mail [39]) was used. This notification mentioned the post of the hacker forum, which we discuss in the next subsection.
- Notification 4 was reached to a network administrator of our university via a domestic CERT on December 17th, 2019.

From the above notifications, it is unveiled that ethical security experts visited our honeypot in addition to malicious visitors.

4) Discussion on attacks in the wild: In Section V, we found evidence of attackers searching for the remote management devices and willingness to manipulate them. This fact raises the question of whether such attacks have been reported as occurring in the wild. To answer this question, we investigated related literature and found the number of reported incidents of ICS is limited due to the critical and confidential nature of these systems. We were not able to find an explicit case where remote management devices were the entry points of the attacks. The closest articles we could find were Japanese media covers in April and May 2018 when network cameras monitoring waterways, rivers, wind power stations, and solar power plants were accessed via their WebUI and some of them could not continue remote monitoring although these articles do not clearly mention if there was ICS behind the remote monitoring system [40], [41]. From the Internet scans, we noticed a portion of the discovered management devices

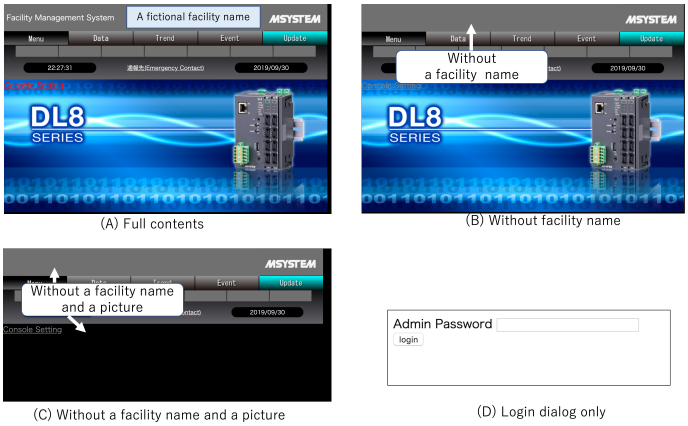

Fig. 14. Types of honeypot WebUI — For (A), (B), and (C), we use a WebUI of DL8, and for (D), we deploy a simple HTML file instead of the WebUI of DL8.

worked with network cameras as it provides additional (visual) status of the monitored facilities. There are several related works: attacks targeting Internet-connected HMIs [42], an attack against a WebUI of an industrial controller [43], PoC code to exploit HMI's WebUI vulnerability [44] and attack observation against WebUI using honeypot [45].

B. Generalizability of the device discovery method

The key intuition behind our detection method is that remote management device UIs contain customized fields. These field distinguish them from more homogeneous UIs of IoT appliances on the one side and more heterogeneous UIs of website on the other side. To detect the presence of customized fields, we do not use the content of these fields and thus we do not require prior knowledge of that content, like facility names or operator names. The technique is content-agnostic and aims to automatically detect these "customized fields" by comparing the WebUI contents of the devices belonging to the same cluster (devices of the same kind) and finding a specific field where each device has unique text. Such text are typically set by the device owners/operators to clearly indicate for themselves what facility they are monitoring. The scan process itself is automated and language-independent and can be applied globally.

In the evaluation phase, where we manually inspect the clusters detected by the scans, we do inspect the content of the custom fields to identify true positives for ICS. Given the automatically extracted texts, we make a final decision whether they indeed indicate ICS facilities. This decision is rather simple and straightforward, as these texts include generic terms indicating the type of facilities such as "water gate", "power plant", and "airport". We believe that one of the reasons why the facility names and types are clearly displayed in their WebUI is for the remote operators to confirm what facilities they are monitoring. This evaluation does imply that using the method in different geographical settings would require some effort to interpret that the content in that specific language is in fact indicative of ICS.

TABLE X
TARGET DEVICES OF PENETRATION TESTS

Device model	Functionalities and use cases	Manufacturer's Web page
SolarView	SolarView is used for monitoring a solar power plant. It can report the status of the solar power plant, such as generated energy and errors of a power conditioner.	https://www.contec.com/products-services/environmental-monitoring/solarview/
DL8	DL8 is used for monitoring general infrastructures such as water facilities and power plants. It has analog and digital inputs and can record the values. Operators can view the recorded values using its WebUI.	https://www.m-system.co.jp/english/products/weblogger/dl8_top.html
Calsos CSDJ	Calsos CSDJ is used for monitoring and managing general infrastructures. Similar to DL8, CSDJ also has analog/digital inputs and data logging functions. In addition, it has a digital output to control facilities.	https://www.necplatforms.co.jp/product/enkaku/index.html

TABLE XI
DISCOVERED ZERO-DAY VULNERABILITIES

Device	Discovered vulnerability	CVSS v3 base score	CVE number
SolarView	OS command injection that allow executing arbitrary commands under privileges of a Web server	6.3	CVE-2021-20658
	Misuse of a hidden Web-based text editor that allows attackers to access arbitrary files	6.3	CVE-2021-20657
	Vulnerable root password that allows privilege escalation	4.6	CVE-2021-20657 and CVE-2021-20658 include this vulnerability
	Not yet disclosed		Not yet disclosed
	Use of old Web server and FTP server that have known vulnerabilities	4.0 (v2 score) 5.0 (v2 score) 9.3 5.0 7.6 (v2 score) 7.5 5.0 (v2 score)	CVE-2011-0762 CVE-2011-4362 CVE-2013-4508 CVE-2013-4559 CVE-2013-4560 CVE-2014-2323 CVE-2014-2324
	Directory listing that allows attackers to obtain directory structure and files in the directories	3.5	CVE-2021-20656
	Reflected cross-site scripting	6.1	CVE-2021-20660
	Improper access control that allows attackers to know/modify a part of device configurations	4.3	CVE-2021-20662
	Improper validation against uploaded files that allows attacker to upload arbitrary files such as a PHP-based backdoor	5.5	CVE-2021-20659
	Directory traversal caused by a file delete function that allows deleting arbitrary files	6.3	CVE-2021-20661
DL8	Improper handling of XML files that is vulnerable for XML bomb	6.5	CVE-2021-20675
	Improper access control that may cause unwanted operations by attackers	4.3	CVE-2021-20676
Calsos CSDJ	Improper access control that allows malicious users to access reports about events	4.3	CVE-2021-20653

TABLE XII
DEVICES DISCOVERED BY OUR SCAN METHOD (DOMESTIC IP ADDRESS RANGES)

We anonymize device model names and manufacturers to avoid the potential harm of misuse. We will provide the detailed device information to interested researchers on a request basis. Note that during the signature generation steps, we identified 23 devices. However, before the final step, device b was removed from the Internet; thus, the number of device b is zero.

Device model name	Manufacturer	Typical use case	# Devices
SolarView Compact	CONTEC CO., LTD	Monitoring of a solar power plant	311
DL8	M-System Co., Ltd.	General-purpose monitoring using analog/digital inputs/outputs	39
Calsos CSDJ	NEC Platforms, Ltd.	General-purpose monitoring using analog inputs/outputs and digital outputs	71
a	α	General-purpose monitoring using analog/digital inputs/outputs	16
b	β	Management of a water or sewer system	0
c	β	Management of a water or sewer system	18
d	γ	Management of a water or sewer system	3
e	δ	Management of a water or sewer system	15
f	ϵ	Monitoring energy usage used in solar power plants, water management system, etc	36
g	ϵ	General-purpose monitoring using analog/digital inputs and analog outputs	22
h	ζ	Monitoring energy usage of a facility	26
i	η	General-purpose monitoring using analog inputs/outputs and digital outputs	141
j	θ	Management of a hydroelectric power plant	13
k	ι	Alive monitoring and rebooting a system	89
l	κ	Monitoring of power usage of a facility or energy generation of a solar power plant	5
m	λ	Monitoring of energy consumption of a facility	2
n	μ	Monitoring of a solar power plant	41
o	ν	Monitoring of a system for landslide	3
p	ξ	Management of a facility such as a water system	3
q	o	Managing of a facility such as a water system, a factory, or a building	9
r	π	Management of a hydroelectric power plant or a water treatment facility	18
s	ρ	Monitoring of a solar power plant	4
t	σ	Monitoring of energy generation of a solar power plant or monitoring of energy consumption of a facility	5

TABLE XIII
DEVICES DISCOVERED BY OUR SCAN METHOD (GLOBAL IP ADDRESS RANGES)

Device model name	Manufacturer	Typical use case	# Devices
A	A	Management of a bat deterrent system	5
B	A	Management of a bat deterrent system	3
C	B	Management of a solar power plant	39
D	Γ	Power management of a facility	1,019
E	Δ	GNSS Receiver	1,204
F	E	Management of a tower lighting system	34
G	Z	Energy monitoring of a facility	1,531
H	H	Monitoring of facility environment (temperature, wind speed, etc)	40

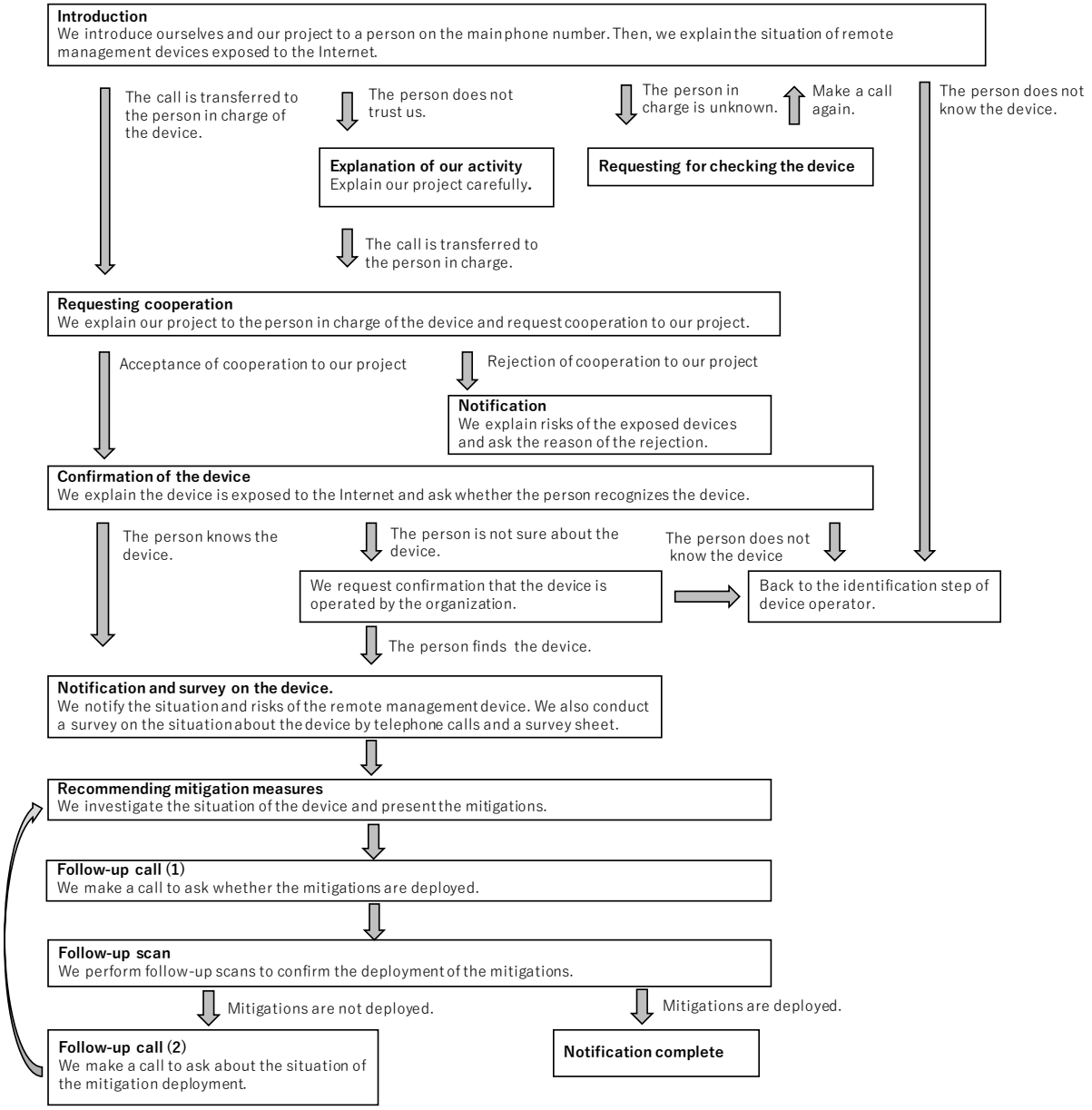

Fig. 15. Notification procedure based on telephone calls