Hardening Honeypots for Industrial Control Systems

Joseph T. Meier
Naval Postgraduate School
joseph.meier@nps.edu

Thuy D. Nguyen
Naval Postgraduate School
tdnguyen@nps.edu

Neil C. Rowe
Naval Postgraduate School
ncrowe@nps.edu

Abstract

Honeypots are computers that collect intelligence about new cyberattacks and malware behavior. To be successful, these decoys must allow attackers to probe a system without compromising data collection. Previously, we developed an industrial control system (ICS) honeypot simulating a small electric-distribution system, but this honeypot was attacked, and its log data was deleted. The current work analyzed the attacks and developed methods to harden the main weaknesses of the public user interface. The hardened honeypot included more robust data collection and logging capabilities, and was deployed in a commercial cloud environment. We observed significant scanning and new attacks, including the well-known BlueKeep exploit and activity related to Russian cyberattacks on Ukraine. Our results showed that the added security controls, monitoring, and logging were more effective in protecting the honeypot's data and event logs.

Keywords: honeypot, industrial control system, RDP attack, logging, cybersecurity

1. Introduction

Attacks on industrial control systems (ICSs) have become common. During the recent conflict between Russia and Ukraine, cyberattacks on ICSs have occurred with major military operations. In April 2022, the Russian hacker group Sandworm launched cyberattacks called Industroyer2 and CaddyWiper against an ICS network controlling electrical substations in Ukraine (Zorz, 2022). Russian cyber actors also used Industroyer in 2015 and 2016 against electric grids in Ukraine to cause widespread blackouts (Cherepanov, 2017). Industroyer could exploit several common ICS protocols including IEC 60870-5-101 (IEC101), IEC 60870-5-104 (IEC104), and IEC 61850. IEC104 traffic is transported over TCP/IP and used for electrical-power substation control and supervision (Clarke & Reynders 2004). Industroyer2, the updated malware Russia used in its most recent (2022) cyberattack, only targets IEC104 devices.

ICSs are high-value targets for attackers and would likely be targeted during a major conflict. Even outside of conflicts, ICSs are probed for weaknesses that could be exploited in the future. Honeypots (decoy systems) are one defensive tool to collect data on attacks and malware used to compromise ICSs. As attackers access the honeypot and move within it, logs provide defenders with attack patterns, entry points, and malware types.

In prior experiments, we developed a honeypot that simulated an ICS for an electric grid (Dougherty, 2020). The honeypot included a Windows-based human-machine interface (HMI) that monitored and controlled the simulated grid. The user interface hosted a Supervisory Control and Data Acquisition (SCADA) program that communicated with the grid over the IEC104 protocol. The honeypot was deployed twice; both times its Windows machine was attacked, and the tools monitoring the honeypot were disabled. Incomplete logs made recreating the attacks impossible.

To avoid future loss of log data, we developed a framework to harden the software configuration of the publicly accessible ICS user interface. Our approach used a robust logging mechanism that recorded security-relevant events in detail and stored them on a separate site. We deployed the hardened honeypot in a cloud environment and ran four experiments. The data collected was analyzed for the tactics that attackers used and to enable fine-tuning the honeypot's monitoring and logging.

2. Threat Model

Attacks against ICS systems can try to change running processes. An example is an attack on a water treatment plant which increased the amount of lye in the water (Cybersecurity and Infrastructure Security Agency, 2021). Attackers exploited a vulnerability in a remote-desktop service running on an Internet-connected engineering station to access its SCADA

The views expressed in this material are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

URI: https://hdl.handle.net/10125/103440
978-0-9981331-6-4

system. Other attacks on ICS have involved ransomware which encrypted data until the operator paid a ransom; an example is the Colonial Pipeline ransomware attack in April 2021 (Parfomak & Jaikaran, 2021). It is believed that initial access was gained using stolen credentials for the virtual private network. Attackers infected computers on the information technology (IT) network with ransomware that demanded money in exchange for the encryption key, forcing the pipeline company to shut down the pipeline for six days to prevent the malware from reaching the ICS network.

Windows 7 was popular and is still being used in control systems. Our honeypot used Windows 10 but was configured to have very lax password management to simulate a poorly configured Windows 7 system.

2.1. Evasion of Logging

Attackers can try to conceal their activity with many evasion techniques (MITRE ATT&CK, 2021). A honeypot is useless if its monitoring can be subverted by an attacker. Tools used for network monitoring attackers such as Wireshark should be carefully protected to avoid losing captured traffic traces. Operating systems log security events such as user logons and logoffs, remote connections, registry modification, and process creation and termination. Attackers can try to disable host-based event logging in general or only for suspicious actions they take.

The security audit policy on Windows systems specifies events to log (Microsoft, 2021a). Permissions for modifying registry keys or files that affect logging can be restricted. It is easy to detect a gap in the data if an attacker erases event logs to conceal their activity. Other countermeasures for log erasure are encrypting log files and logging remotely.

2.2. Remote Desktop Protocol

Although virtual-private-network technology allows secure remote access, the Remote Desktop Protocol (RDP) less-secure desktop-sharing protocol (Microsoft, 2021b) is common on Microsoft Windows systems. RDP enables network administrators to remotely control computers on a network, but it can also allow users to access unique resources available at a workstation. The increase in employees working from home due to COVID-19 has encouraged such use.

Once an RDP connection is established, data is transferred between server and client. Typically, the server sends data representing its desktop screen to the client machine. The client displays the screen image and sends keystrokes and cursor movements to the server for processing. During an active RDP connection, dynamic and static virtual channels are established between client and server (Microsoft, 2022). Both the server and client must have channel managers to initialize and maintain RDP sessions.

RDP is useful for industrial control systems because it enables remote supervision. RDP also offers many features that can be exploited by attacks. BlueKeep and DejaBlue are well-known exploits (MITRE, 2019a) which target the virtual channels. BlueKeep starts with the client requesting an internal-communications channel MS_T120. Remote connections requesting its creation are not legitimate and distinctive to BlueKeep. DejaBlue exploits a vulnerability in decompression of data sent over the DRDYNVC channel. Sending a carefully crafted string over this channel overflows the heap and gives the attacker control. DRDYNVC is commonly used in legitimate RDP connections, so its use is an insufficient indicator of DejaBlue.

3. Honeypots and Related Work

Honeypots are systems intended to deceive attackers that they are legitimate systems (Provos, 2004). They can provide a high, medium, or low level of interaction with users (Franco et al. 2021).

3.1. ICS Honeypots

ICS honeypots simulate services for an industrial process. ICS honeypots are more complex than other honeypots because they must imitate the changing states of a physical process. Low-interaction ICS honeypots can simulate a service for communicating with a programmable logic controller (PLC) (Franco et al., 2021). High-interaction ICS honeypots try to accurately simulate physical devices found in ICS, such as a water pump or electric switch, and may have user interfaces to control them and show real-time sensor data.

An early honeypot for ICS systems was the SCADA HoneyNet Project (Pothamsetty & Franz, 2004) which simulated industrial processes like SCADA and PLCs. One implementation was deployed in locations around the world using Amazon's cloud environment (Serbanescu et al., 2015) and simulated the Modbus and IEC104 ICS protocols.

Navarro et al. (2018) built a high-interaction SCADA honeypot that simulated a water-treatment plant, a user interface, and a monitoring system. Another project simulated a small industrial prototyping company with several types of programmable logic controllers, an interface, a firewall, and a file server (Hilt et al., 2020). They observed two ransomware attacks and cryptocurrency mining activity. Another group deployed honeypots on cloud platforms around the world and compared their data (Kelly et al. 2021).

One of the honeypots used RDPY (Citronneur, 2020), a Python implementation of the RDP remote-desktop protocol. They found that the most common attacks were against desktop-sharing services.

3.2. Previous Conpot and Gridpot Work

Conpot is a low-interaction ICS honeypot that emulates several services common in ICSs including IEC104, HTTP, Modbus, and S7Comm (Conpot.org, n.d.). GridPot is a medium-interaction honeypot that includes a high-fidelity simulation of a power-distribution grid, GridLab-D (Sk4ld, 2015). It uses a modified Conpot that interfaces with GridLab-D and can simulate switches, transformers, regulators, and other typical parts of an electric grid.

We used the GridPot and Conpot combination in previous work by our group. We had improved GridPot to translate messages sent to its IEC104 server into variables understood by GridLab-D simulation (Dougherty, 2020). A separate workstation running on a Windows 8 operating system presented a graphical user interface implemented with IndigoSCADA (Enscada, 2021). The RDP protocol was enabled on the Windows 8 machine so remote attackers could manipulate the simulation.

The honeypot was deployed and compromised twice. Wireshark on the Windows machine was disabled by the attackers, which made reconstructing the attacks difficult. Event logs on the Windows machine and some packet captures from Wireshark gave only a few clues to what happened.

4. Design of A Hardened Honeypot

To lure attackers, a good honeypot should provide a publicly accessible site with a user interface that appears to monitor and control something important. Monitoring this interface should be stealthy to not raise suspicion about this system being a honeypot. Furthermore, the data collected should be protected from modification and deletion.

Dougherty's honeypot provided us with a start. However, attackers had quickly disabled its logging by exploiting the Windows 8 machine through the RDP protocol. Although the only accessible account had minimal privileges, the attacker could exploit publicized unpatched vulnerabilities or misconfiguration to gain administrative privileges. Wireshark ran in the administrative account, and attackers could stop it.

Our solution was to run Windows in a virtual machine (VM) and to separately run Wireshark packet capture and logging on Linux (Figure 1). Packet capture and logging could not be stopped unless the attacker exploited a vulnerability to escape the VM. We implemented our design with the DigitalOcean cloud service to distance it from our testbed. Our honeypot ran on three DigitalOcean Linux VMs, which DigitalOcean refers as "droplets" (DigitalOcean, 2020). These were the User-Interface Droplet, the GridPot Droplet, and the Logging Droplet. Each droplet had a public Internet interface, and an interface for communications between VMs in a private network that DigitalOcean calls a "virtual private cloud" (VPC) (DigitalOcean, 2020).

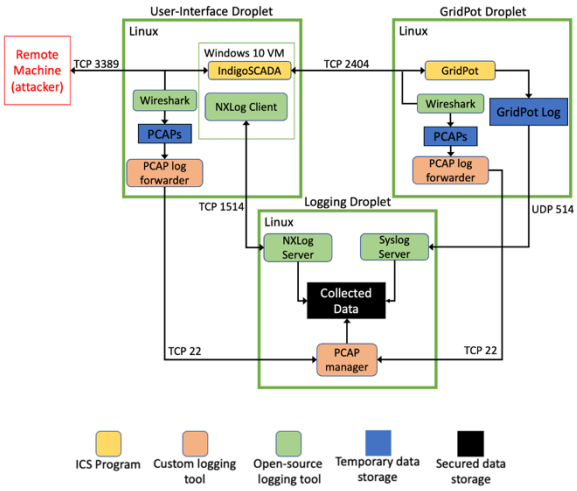

Figure 1. Hardened Honeypot Design

4.1. The User-Interface Droplet

The User-Interface Droplet hosted a Windows 10 VM running the IndigoSCADA program that attackers can use to manipulate the simulated electric grid. We set the DigitalOcean firewall to only allow network traffic to RDP port 3389 on the public interface and ports 22, 2404, 1514, and 12345 on the private interface. The Windows VM used network-address translation, so traffic to port 3389 on the external interface was forwarded to port 3389 of the VM.

The Windows 10 VM had an account with administrative privileges for configuring the workstation, and an account for the SCADA interface which was accessible by the RDP protocol. We disabled remote logon to the administrator account to impede attackers from tampering with the event logs. RDP is commonly accessed over port 3389, so attackers could scan for it to discover the service.

Remote logons to the Guest account were disabled between Windows 8 and Windows 10 for security reasons. To make our honeypot as vulnerable as Dougherty's, we created a user account with a commonly used account name and a blank password. We also disabled network-level authentication so attackers would see the logon screen without first

entering credentials, which encouraged attackers to access the user account. To protect the Windows event logs, we configured NXLog to forward the event logs from the Windows VM to the Logging Droplet in JSON format and send log files every 100 MB. Wireshark ran on the Linux host outside the VM and collected traffic on the public interface of the User-Interface Droplet. Wireshark saved PCAP files every 20MB to a directory; files in the directory were automatically forwarded to the Logging Droplet on port 12345.

4.2. The GridPot Droplet

The GridPot Droplet ran the GridPot software with GridLab-D simulating an IEEE 13-node model with houses. Experiments 1-3 blocked traffic to the HTTP server because we wanted RDP to be the only access to our honeypot. In Experiment 4 we allowed traffic to the HTTP server to advertise our honeypot as an ICS. The HTTP server displayed limited information about the electric grid simulated by GridPot, revealed the public IP address of the User-Interface Droplet, and directed users to access the Windows machine using RDP.

GridPot's IEC104 server listened on TCP port 2404 on its private interface. This was the only public interface to the GridPot Droplet. The SSH secure-shell protocol was also enabled, but it would only accept connections by administrators. Wireshark monitored the private interface, and when PCAP storage reached a limit of 20MB, it was forwarded to the Logging Droplet. For further backup, we also directed GridPot to forward its logs to the Logging Droplet using the syslog protocol over UDP port 514 on its private network interface.

4.3. The Logging Droplet

The Logging Droplet collected packets and logs from all data sources in our honeypot. We also used a cloud-based storage service to back up data of the Logging Droplet. On the private interface, a syslog server listened on port 514 for incoming logs from GridPot, and an NXLog server listened on port 1514 for event logs from the Windows machine running on the User-Interface Droplet. The PCAP manager listened on port 12345 of the private interface for notifications that a new PCAP file was ready to be transferred. When a notification was received, it started an SSH session to copy the PCAP file from the machine that sent the notification. As recommended by NIST (National Institute of Standards and Technology, 2015), an ICS that uses a demilitarized zone (DMZ) typically has a data collector in the control network and a historian server in the DMZ. Our logging server served as the data collector for our honeypot.

5. Experiments and Analysis Method

We ran four experiments with live attackers and analyzed each to determine the tactics the attackers used. We adjusted the logging after each experiment to better capture the data to recreate the attacks.

5.1. Experiment Description

Experiment 1 started November 24, 2021 and collected data for 39 days. We wanted to see if cyberattacks would be more likely during this period since many employees, including network-security personnel, take time off during the holidays. During Experiment 1, the RDP certificate on our Windows machine expired. After that, all RDP sessions captured in the PCAP data could not be decrypted, so we attempted a different method for replaying RDP sessions in Experiment 2.

For Experiment 2, we modified the user interface to use the RDP monitoring tool PyRDP in "man-in-the-middle" mode, which could capture and replay RDP sessions. Experiment 2 ran 25 days from January 25, 2022. It was stopped when we discovered that the way PyRDP handled RDP packets caused the network scanner NMAP to consider the RDP sessions as ungenuine. Afterward, we passively captured RDP network traffic using Wireshark, and then converted the RDP sessions to MP4 files using PyRDP's convert tool.

Experiment 3 ran 24 days from February 19, 2022. We reverted to the configuration in Experiment 1. We renewed the RDP certificate and confirmed that it would not expire during the experiment. To generate interest in the honeypot, we changed the name of the Windows workstation to "HMI-OPERATOR." The name used in Experiments 1 and 2 was DESKTOP-10N1FB, the default computer name for Dell systems. When the system was scanned, the RDP certificate showed this name to suggest it is part of an ICS system. The IndigoSCADA program was also set to automatically run when a user logged in, to make it look more like an ICS system.

Experiment 4 ran 29 days from March 15, 2022. We changed the logging mechanisms significantly and gave the user interface a new look. We also moved the user-interface machine to a different IP address so it would appear to be a new system. The computer name was changed to SGEUtilities. We modified the Windows machine's audit policy for process creation to record the command line for any process created. This would show any options, flags, or arguments passed to a process. We also enabled Script Block Logging for PowerShell to record any script run in it, and used the Windows logging service Sysmon. We changed Sysmon's default configuration file to filter out irrelevant log entries. We

enabled the GridPot's HTTP server which, when accessed, returned a Web page that refreshed every two seconds with updated readings from GridLab-D. To claim that the system was for an electric-distribution company, the page included a message that directed employees to log on to the Windows user interface with RDP at a specific IP address.

5.2. Data Analysis Method

Scanning events were filtered out first from the logs and packet captures. We defined scanning as one-time connections to ports, i.e., no subsequent activity was observed. Event ID 140 records when an RDP logon fails. Event ID 4625 complements 140 by recording the username of the failed logon attempt. Also, any packets directed to the RDP listening port 3389 that did not result in an RDP connection could be considered scanning.

To find successful logins to the Windows machine, we searched the event logs for event ID 4624, which indicates a user login using RDP, and logoff event 4634. We correlated these events with the PCAP files. We used the RDP server key to decrypt the traffic and convert it to an MP4 video file using the PyRDP tool. Other events that we identified in the logs were process creation, registry modification, security (Windows Defender) alerts, changes to logging, and network activity. Event logs could recreate actions unobservable in the RDP video playback such as installation and running of a malicious backdoor on the system.

We analyzed the GridPot logs for IEC104 traffic between the Windows user interface and GridPot. When the IndigoSCADA program executes, GridPot sends IEC104 messages that update the values of the electrical components simulated by GridLab-D. Any IEC104 traffic would indicate that an attacker has logged into the Windows workstation and started IndigoSCADA.

6. Results and Discussion

We examined the collected data for four types of events: RDP port scanning, login attempts, post-login activities, and malicious actions in the Windows 10 VM.

6.1. RDP Port Scanning

Port scanning of our honeypot in Experiment 1 started only ten minutes after the DigitalOcean firewall opened port 3389. We did not advertise our IP address in Experiments 1-3. We only exposed our IP address on GridPot's status Web page in Experiment 4. However, the PCAPs collected in Experiment 1 had IP addresses associated with the Censys scanner, indicating that bots looking for publicly-accessible services found us. We also discovered IP addresses belonging to an IP address block assigned to an alleged Russian organization named Alexander Valerevich Mokhonko. These addresses were used between November 25 and December 1, 2021, and we surmised IP hopping was performed by bots to probe our honeypot.

Figure 2 shows the distribution of the reported countries of origin across all four experiments. Experiments 1-3 had somewhat random distributions, with the most-common country around a third of the total scans observed. Experiment 4 was different in that 67% of scanning originated from IP addresses that appear to be from Russia, which could relate to the ongoing conflict in Ukraine. It may have been preparation for an attack on Ukraine's electric grids that targeted Windows machines of ICS systems. Our honeypot emulated the same kind of IEC104 systems that were targeted in the Ukrainian power-grid attack in April 2022 (Zorz, 2022).

The number of TCP connections to port 3389 are 1,046,198, 67,930, 1,056,909, and 986,001 for Experiments 1-4, respectively. It appears that use of the PyRDP tool makes a site unattractive to attackers since an NMAP scan with PyRDP in the middle failed with the error message "3389/tcp open ms-wbt-server? 1 service unrecognized despite returning data."

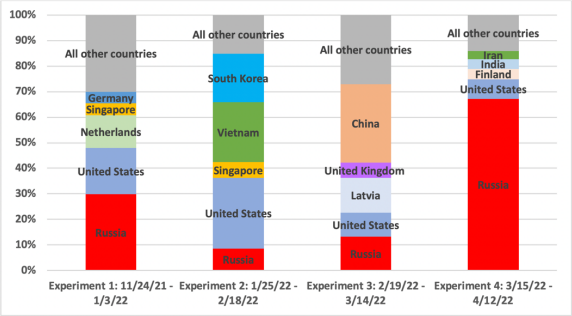

Figure 2. Percentage of Scanning by Country

In Experiment 2, we observed attempts to connect to the MS_T120 channel during the channel-connection phase of the RDP sequence. BlueKeep showed up in all four experiments, but in Experiment 2 it was easier to find because PyRDP logged MS_T120 connection attempts as possible BlueKeep exploits (Yen, 2019). We had to manually examine PCAPs from the other three experiments to identify BlueKeep scans. Since Windows 10 is not vulnerable to BlueKeep, the attempts were likely automated and only directed at our honeypot because port 3389 was exposed.

We also analyzed the local time when attackers started an RDP session with our honeypot since that could suggest the type of attacker. Logins during typical working hours would more likely come from a

professional organization such as a state-sponsored cyber group. All local times of the RDP sessions are in Table 1. Experiment 2 data was excluded from the Table 1 because no logins were observed. We could not discern any patterns that would indicate professional activity in our data, as the local times of the RDP sessions were evenly spread throughout the day. Furthermore, we could not be certain that the attackers did not use techniques to make their IP address appear to be in a location other than their true one.

Table 1. Local Times of the Alleged Countries that Started RDP Sessions

Experiment 1			Experiment 3			Experiment 4		
Time (PST)	Alleged Source Country	Alleged Source Local Time	Time (PST)	Alleged Source Country	Alleged Source Local Time	Time (PST)	Alleged Source Country	Alleged Source Local Time
12/27/21 9:15	Russia	19:15	2/20/22 3:48	Norway	12:48	3/18/22 10:02	Ukraine	20:02
12/27/21 10:28	Germany	19:28	2/20/22 5:19	Norway	14:19	3/18/22 22:38	Iran	10:08
12/28/21 7:19	United States	7:19	2/21/22 0:19	Germany	9:19	3/18/22 22:38	Germany	7:38
12/28/21 9:42	United States	9:42	2/23/22 0:38	Russia	10:38	3/18/22 22:41	Germany	7:41
12/28/21 14:07	United States	14:07	2/27/22 0:01	Russia	10:01	3/19/22 10:21	Ukraine	20:21
12/28/21 16:52	Belize	17:52	3/7/22 15:39	Russia	1:39	3/19/22 10:52	Ukraine	20:52
12/29/21 2:56	Ukraine	12:56	3/7/22 17:44	Latvia	3:44	3/20/22 8:45	Russia	18:45
12/29/21 16:52	United States	16:52	3/8/22 3:55	Bulgaria	13:55	3/21/22 8:53	Ukraine	18:53
12/29/21 17:12	United States	17:12				3/21/22 9:33	France	18:33
12/29/21 17:43	United States	17:43				3/21/22 9:36	Iran	21:06
12/30/21 11:06	United States	11:06				3/22/22 9:29	Russia	19:29
12/30/21 13:41	Russia	1:41				3/22/22 10:20	Germany	18:20
12/31/21 12:51	United States	12:51				3/22/22 11:07	Ukraine	21:07
						3/23/22 8:52	Ukraine	18:52
						3/24/22 9:01	Ukraine	19:01
						3/27/22 5:25	Myanmar	18:55
						3/27/22 7:42	Iran	19:42
						3/29/22 8:09	Ukraine	18:09
						4/1/22 11:52	Vietnam	1:52
						4/1/22 14:54	Iran	2:24

6.2. Login Attempts

After scanning the RDP port, attackers guessed account names on the Windows machine. Table 2 shows the frequency of interaction in terms of the top ten usernames guessed in brute-force attacks. No login attempts were observed in Experiment 2.

Most guessed account names were for variant names of the ADMINISTRATOR account. Also popular was "WHATUPTIME.COM", the default username for a version of Windows created specifically to run on DigitalOcean's platforms. This indicates that attackers guessed our honeypot was a Windows machine owned by DigitalOcean, yet still attacked it. Similarly, attackers still tried to attack a previous instance of our honeypot on DigitalOcean even though the Shodan Honeyscore tool reported it as "highly likely" to be a honeypot. In Experiment 3, after changing the name of the Windows VM to "HMI-OPERATOR", we saw "HMI-OPERATOR", "HMI", and "OPERATOR" among the top ten most-guessed usernames, which could mean attackers adjusted their guesses based on feedback. Event ID 4625 ("Account failed to logon") logged guessed usernames and their authentication methods.

Table 2. Top Ten Most-guessed Usernames

Experiment 1		Experiment 3		Experiment 4	
Count observed	Username	Count observed	Username	Count observed	Username
396882	ADMINISTRATOR	511486	ADMINISTRATOR	193243	ADMINISTRATOR
32306	administrator	156302	WHATUPTIME.COM	44622	administrator
27803	ADMIN	59417	Administrator	33276	Administrator
11247	USER	44569	ADMIN	22049	WHATUPTIME.COM
8957	WHATUPTIME.COM	9339	HMI-OPERATOR	14642	ADMIN
2280	Administrator	7064	Administrador	9151	USER
2123	TINHOCTHUCHANH	6461	OPERATOR	5586	Administrador
1707	MINER	6209	HMI	5188	Administrateur
1677	CHIA	6184	Administrateur	1534	null
1529	TEST	4858	USER	1401	WINADMIN

The username "TINHOCTHUCHANH" was guessed 2123 times. It is a Vietnamese phrase that, according to Google Translate, means "practical informatics." Since many Vietnamese IP addresses scanned our honeypot, we filtered our data to find the countries associated with the IP addresses that guessed that username. We discovered that no Vietnam-based scanner used it.

6.3. Post-login Activities

At least 13 successful logins to the user interface occurred in Experiment 1. Also, user activity was reported in the event logs without corresponding login events, so either some logins were not logged or were deleted by the attackers. Table 3 shows the times of the recorded logins and durations of the RDP sessions.

Noticeably, all logins in the 39 days occurred between Christmas and New Year's Day when many employees take leave from their jobs. Most were allegedly from an IP address within the United States. The remainder were shorter RDP sessions lasting two minutes or less. Only one interaction with the IndigoSCADA program occurred, when an icon on the desktop was clicked to start it. This was marked in the Windows event log with Event ID 4688. The attacker did not interact further with the program beyond launching it. Interactions marked with "Firefox" in the table related to an attack involving the Firefox Web browser. We could not determine the role of Firefox in these attacks because we could not decrypt RDP sessions captured in the PCAP data in Experiment 1 when the RDP certificate expired. "Recon" refers to actions taken by the attacker to gather information about their target, such as looking for active services, IP addresses, or other machines in the same network as the target.

Table 3. Experiment 1 - Successful Logins

Time (PST)	Length of RDP Session (minutes)	Alleged Source IP Address	Alleged Source Country	Interaction if any	Alleged Source Local Time
12/27/21 9:15	<1	87.251.64.137	Russia	Recon	19:15
12/27/21 10:28	2	146.0.40.37	Germany	Firefox	19:28
12/28/21 7:19	34	45.130.83.24	United States	Firefox	07:19
12/28/21 9:42	17	45.130.83.150	United States	Firefox	09:42
12/28/21 14:07	43	45.130.83.24	United States	Firefox	14:07
12/28/21 16:52	<1	45.227.254.118	Belize		17:52
12/29/21 2:56	1	77.83.36.32	Ukraine	SCADA	12:56
12/29/21 16:52	8	154.6.16.155	United States		16:52
12/29/21 17:12	21	154.6.16.155	United States	Firefox	17:12
12/29/21 17:43	31	198.255.5.170	United States		17:43
12/30/21 11:06	3	45.130.83.150	United States	Firefox	11:06
12/30/21 13:41	1	185.191.32.160	Russia		01:41
12/31/21 12:51	46	45.130.83.145	United States	Firefox	12:51

More logins occurred during Experiment 4 (Table 4); most were short (1 minute or less).

Table 4. Experiment 4 - Successful Logins

Time (PST)	Duration of RDP session (minutes)	Alleged Source IP Address	Alleged Source Country	Interaction if any	Alleged Source Local Time
3/18/22 10:02	<1	31.43.185.9	Ukraine		20:02
3/18/22 22:38	<1	185.56.150.158	Germany		07:38
3/18/22 22:38	2	5.114.247.195	Iran		10:08
3/18/22 22:41	<1	185.56.150.158	Germany		07:41
3/19/22 10:21	<1	31.43.185.9	Ukraine		20:21
3/19/22 10:52	<1	31.43.185.9	Ukraine		20:52
3/20/22 8:45	<1	94.232.42.186	Russia		18:45
3/21/22 8:53	<1	31.43.185.9	Ukraine		18:53
3/21/22 9:33	<1	51.195.189.7	France		18:33
3/21/22 9:36	1	5.120.227.231	Iran	SCADA	21:06
3/22/22 9:29	<1	87.251.64.26	Russia		19:29
3/22/22 10:20	6	159.242.234.121	Germany	Recon	18:20
3/22/22 11:07	<1	31.43.185.9	Ukraine		21:07
3/23/22 8:52	1	31.43.185.9	Ukraine		18:52
3/24/22 9:01	<1	31.43.185.9	Ukraine		19:01
3/27/22 5:25	<1	43.245.46.142	Myanmar		18:55
3/27/22 7:42	2	5.119.130.106	Iran		19:42
3/29/22 8:09	1	31.43.185.9	Ukraine		18:09
4/1/22 11:52	<1	58.186.205.49	Vietnam		01:52
4/1/22 14:54	1	5.190.75.174	Iran		02:24

Only one RDP session lasted more than two minutes, that on March 22, 2022 at 10:20AM PST. Unique to Experiment 4 was the high number of logins from IP addresses claiming to originate from Ukraine, all of which began after 6:00PM local time. These logins may have related to the then-current cyber conflict between Ukraine and Russia. Two factors could have caused greater activity on our honeypot. First, the webpage from GridPot's HTTP server revealed the IP address of our SCADA interface, and that it could be accessed using RDP. Second, "User" was the username for the Windows machine, a popular username in previous experiments.

In Experiment 3, despite much scanning, attackers interacted very little with the interface. Only eight logins were successful over RDP; all but one was less than a minute long. The long session came from Norway and lasted for 19 minutes, but the attacker did not start the SCADA process. The other RDP interactions were likely automated to determine if it was possible to log in. We concluded that attackers had little interest in our honeypot in Experiment 3, perhaps because they remembered PyRDP from Experiment 2 or recognized that the same IP address from Experiment 1, as supported by the fact that Experiment 4 had more post-login activities after we changed the IP address.

6.4. Malicious User Actions

In Experiment 1, the first observed user activity in the Windows 10 VM occurred on December 26 between 3:53AM and 3:56AM PST. A program named Advanced_IP_Scanner.exe was executed from the C:\Users\<username>\AppData folder, but no other events indicated what it did. No log events showed how the program got on the machine despite our modification of the logging configuration. No RDP traffic was recorded during this time, which might mean an attacker logged on at an even earlier time and dropped a program or script to execute the scanner. This attacker likely evaded our logging system by being careful not to trigger certain events, erasing log entries pertaining to the attack before the logs were forwarded to the logging server, or changing the logging policy. However, there was no evidence that the attacker found the logging system, or the logging policy was modified.

The next attack happened the following day at 10:28AM PST. We recreated the attack with data from the Windows event logs and PCAPs. A summary of the chain of events is in Figure 3. An RDP session began, and the attacker dropped a program named c.exe on the desktop of the Windows machine which started a chain of events. The program made an HTTP GET request to codeproject.com to download a Firefox installer program. It then connected to ipinfo.io/json and icanhazip.com to apparently track machines already infected. After c.exe received responses from both, the Firefox installer downloaded the Firefox web browser, and the attacker logged off the machine.

Over the next four days, similar RDP sessions were started from an IP address in the United States. They all used the Firefox browser to access travel websites including mytrip.com, kayak.com, and destinationholidays.com. These sites may have been hosting adware to which the version of Firefox was vulnerable. After Experiment 1 ended, we took a snapshot of the Windows machine for analysis and opened the Firefox browser while running Process Explorer, a Sysinternals tool, to see if it spawned other processes, but we did not observe anything suspicious. VirusTotal.com identified the SHA256 hash as a

legitimate Firefox browser. Further analysis is needed to determine the role of the Firefox browser in this attack.

Although more logins occurred during Experiment 4, we did not observe as much interaction with the Windows VM as in Experiment 1. In one RDP session, on March 21, the attacker opened the IndigoSCADA program but did not interact with it. An attacker apparently performed reconnaissance on March 22 and, after downloading the Advanced Port Scanner tool using Microsoft Edge, did a limited scan of the local network of the VM. They first scanned to find other running services not remotely accessible. Then they sequentially scanned the subnet starting at the private address 10.0.2.2, stopping at 10.0.2.4. With network-address translation (NAT), VirtualBox's gateway router defaults to 10.0.2.2 and the VM is assigned 10.0.2.15. The IP addresses that the attacker scanned indicate that they may have known this was a VM running in VirtualBox. Some ports scanned were for common services, like SSH (port 22), SNMP (port 161), and IPP (631). Other scanning looked for other machines that were running RDP (port 3389) or VNC (port 5901). Finally, they scanned port 80 of IP address 188.40.30.100, an IP address in Germany that hosts the advanced-port-scanner.com domain. This behavior indicates that the attacker was trying to move laterally in the network.

CPU use of the User-Interface Droplet suddenly increased in the middle of Experiment 4. During the previous experiments, the CPU use was around 30%, but on April 1 it jumped to about 53% and remained at that level until April 11. The sudden increase coincided with the last login to the Windows machine. No user activity in the event logs explained such a large jump in CPU use. On April 11 the Windows machine crashed and reverted to the snapshot taken just before Experiment 4. We have been unable to confirm the cause of the crash, but we have several hypotheses. Figure 4 shows the suspicious events in Experiment 4.

One hypothesis is that an attacker logged in to the Windows VM on April 1, installed malware that performed CPU-intensive tasks such as cryptocurrency mining, and disabled remote access for everyone but themselves. They would need to tamper with the logging policies to prevent their actions from being logged. Cryptocurrency mining could explain the large spike in CPU use on April 1. It appears they also disabled remote access to the Windows VM to avoid intervention, which would explain why logins suddenly stopped.

Another hypothesis is that the crash was related to Industroyer2. The attack ostensibly targeted Windows machines in an ICS network owned by a Ukrainian electric-grid operator, but also spread to other countries. The timelines for the Russian attacks match what we observed on our honeypot.

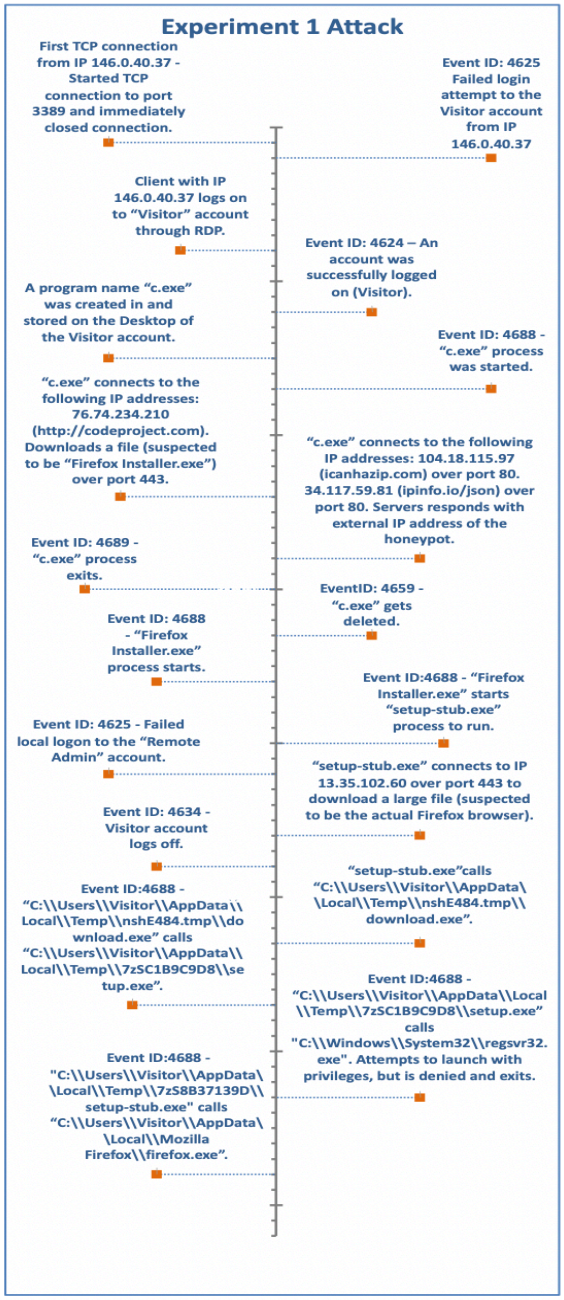

Figure 3. Experiment 1 - Attack Timeline

Once the Russian attack on Ukraine's power grid was thwarted, our attackers may have reverted our Windows VM to an earlier snapshot to conceal their activity. The scan from the Advanced Port Scanner tool suggests that the attacker determined our Windows machine was running in VirtualBox, since the IP addresses they scanned were default settings for using NAT in VirtualBox. An advanced attacker could have used this information to exploit the VirtualBox hypervisor to do this. After our Windows VM reverted to an earlier snapshot, our honeypot was no longer

remotely accessible, since the snapshot was created before the user account that could use RDP was created.

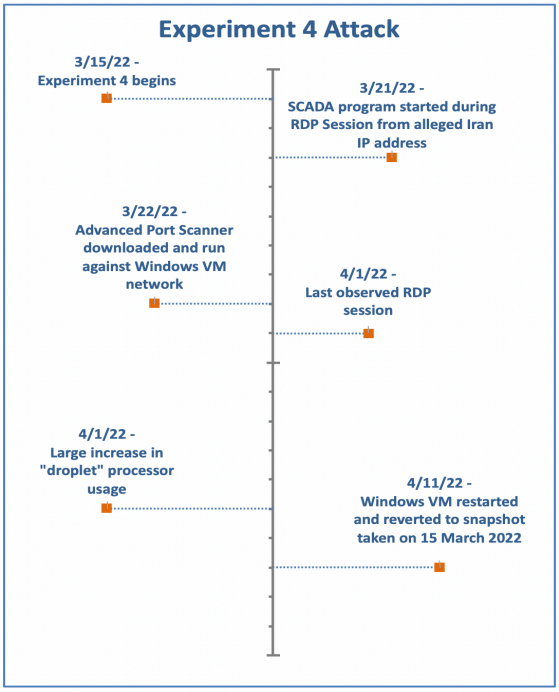

Figure 4. Experiment 4 - Attack Timeline
(Vertical spacing is not to scale)

In Experiment 4, we changed GridPot's HTTP server to display a webpage that contained the IP address of our User-Interface Droplet and instructed users to access it using RDP. We wanted to determine if the information displayed on the webpage increased traffic to the User-Interface Droplet. We extracted all IP addresses that sent an HTTP GET request to retrieve the GridPot interface webpage and compared them against IP addresses that started a TCP connection to the RDP listening port of the User-Interface Droplet. Of the approximately 1700 unique IP addresses that established connections to either of the two machines, only 21 addresses visited both GridPot's HTTP server and the User-Interface Droplet. Further analysis showed that 15 of the 21 IP addresses visited the HTTP server first. Time between visits ranged from 5 hours to 15 days. Also, no IP address tried to make an RDP connection with the Windows machine. They only started TCP connections, which indicates that they were scanning the machine to find an open port. Evidence was insufficient to conclude that enabling the HTTP server increased traffic to the User-Interface Droplet. However, it is possible that once an attacker gathered information from the HTTP server, they used a machine with a different IP address to connect to the User-Interface Droplet, and we could not correlate those two connections.

Future work will investigate whether suspicious artifacts left behind in Experiment 1 like the c.exe executable would inform other attackers that our Windows system was a realistic vulnerable target and that it was already compromised.

We examined the event logs saved on the Logging Droplet for clues about the sudden increase in processor usage. Sysmon should have logged any file-creation or process-creation events that are resource-intensive. We found no occurrence of the Event IDs 4 (Sysmon service state changed), 16 (service configuration change), 23 (file delete archived), 25 (process tampering), or 26 (file delete detected). Event ID 5 (process terminated) was logged once when an updater program was terminated. There were occurrences of Event ID 1 (process create) and Event ID 11 (file create) but nothing suspicious stood out. We also examined the PowerShell Script Block logs for evidence of a malicious PowerShell script and found nothing suspicious there either. We surmise that the attacks were "living-off-the-land" attacks in which attackers took advantage of legitimate services to carry out the malicious actions.

6.5. Protection of Collected Data

Our honeypot's logging system worked as intended. Table 6 shows the total size of both the Event Log and PCAP datasets for each experiment.

Table 6. Dataset Size Comparison across All Experiments

	File Size (GB)		
Experiment	**Event Log**	**PCAP**	**Experiment Duration (Days)**
1	26.08	11.52	39
2	3.85	6.34	24
3	19.52	5.69	24
4	27.86	12.21	28

Experiment 2 produced the least data since it lacked logins to the Windows machine. Experiment 4's increase in data was because with Sysmon running on the Windows machine, more events were being logged, as well due to, allegedly, increased Russian cyber activities related to Ukraine.

During Experiment 4, if NXLog did not back up the event logs from the Windows machine to our logging server, we would have lost all event-log data when the virtual machine reverted to a snapshot taken before Experiment 4 began. Furthermore, no PCAP data was lost either; Wireshark was not interrupted during any of the four experiments and all PCAP files were forwarded to the logging server.

7. Conclusions

Our honeypot was sufficiently realistic to entice professional attackers to interact with it. We learned from each experiment, and logging was increasingly hardened to withstand attacks and still provide enough data to determine attack patterns and behaviors. Despite our efforts to draw attention to the ICS functions of our honeypot, intruders were more interested in attacking the Windows machine than the ICS functions. However, the close correlation of some attacks we observed in April 2022 with concurrent Russian Industroyer2 cyberattacks on ICSs in Ukraine, which used the same IEC 104 protocol, is unlikely to be accidental.

We recommend making an ICS-honeypot environment more restrictive so attackers are more likely to interact with the ICS functions of the honeypot rather than the Windows machine itself. Windows has a "kiosk mode" that only allows one application to be run when logged in. Running the Windows machine in this mode could increase the likelihood of an attacker interacting with the SCADA interface.

8. References

Cherepanov, A. (2017). WIN32/INDUSTROYER: A new threat for industrial control systems. WeLiveSecurity. https://www.welivesecurity.com/wp-content/uploads/2017/06/Win32_Industroyer.pdf

Citronneur (2020). RDPY. [Computer software]. Github. https://github.com/citronneur/rdpy

Clarke, G., & Reynders, D. (2004). Practical modern SCADA protocols: Dnp3, 60870. 5 and related systems. Elsevier Science & Technology.

Conpot.org (n.d.). Conpot ICS/SCADA Honeypot. http://conpot.org

Cybersecurity and Infrastructure Security Agency (2021). Compromise of U.S. Water Treatment Facility. https://www.cisa.gov/uscert/sites/default/files/publications/AA21-042A_Joint_Cybersecurity_Advisory_Compromise_of_U.S._Drinking_Treatment_Facility.pdf

DigitalOcean (2020). VPC Quickstart. https://docs.digitalocean.com/products/networking/vpc/quickstart/

Dougherty, J. (2020). Evasion of honeypot detection mechanisms through improved interactivity of ICS-based systems. [Master's Thesis, Naval Postgraduate School]. http://hdl.handle.net/10945/66065

Enscada (2021). IndigoSCADA [Computer software]. Github. https://github.com/enscada/IndigoSCADA

Franco, J., Aris A., Canberk, Berk. & Uluagac, A. (2021). A Survey of Honeypots and Honeynets for Internet of Things, Industrial Internet of Things, and Cyber-Physical Systems. https://arxiv.org/pdf/2108.02287.pdf

Hilt, S., Maggi, F., Perine, C., Remorin, L., Rösler, M, & Vosseler, R. (2020). Caught in the Act: Running a Realistic Factory Honeypot to Capture Real Threats. Trend Micro Research. https://documents.trendmicro.com/assets/white_papers/wp-caught-in-the-act-running-a-realistic-factory-honeypot-to-capture-real-threats.pdf

Kelly, C., Pitropakis, N., Mylonas, A., McKeown, S., & Buchanan, W. J. (2021). A comparative analysis of honeypots on different cloud platforms. Sensors, 21(7), 2433. http://dx.doi.org/10.3390/s21072433

Microsoft (2021a). Audit Policy. https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/audit-policy

Microsoft (2021b). Understanding the Remote Desktop Protocol (RDP). https://docs.microsoft.com/en-us/troubleshoot/windows-server/remote/understanding-remote-desktop-protocol

Microsoft (2022). Microsoft NTLM. https://docs.microsoft.com/en-us/windows/win32/secauthn/microsoft-ntlm

MITRE (2019a). CVE -CVE-2019-0708. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-0708

MITRE ATT&CK (2021). Defense Evasion. ATT&CK Matrix for Enterprise. https://attack.mitre.org/tactics/TA0005/

National Institute of Standards and Technology (2015) Guide to Industrial Control Systems (ICS) Security. Federal Information Processing Standards Publications (FIPS PUBS) 800-82, Revision 2

Navarro, Ó., Balbastre, S. & Beyer, S. (2019). Gathering Intelligence Through Realistic Industrial Control System Honeypots. Critical Information Infrastructures Security. CRITIS 2018. https://doi-org.libproxy.nps.edu/10.1007/978-3-030-05849-4_11

Parfomak, P. & Jaikaran, C. (2021) Colonial Pipeline: The DarkSide Strikes. Congressional Research Service. https://crsreports.congress.gov/product/pdf/IN/IN11667

Pothamsetty, V. & Franz, M. (2004). SCADA HoneyNet Project: Building Honeypots for Industrial Networks. http://scadahoneynet.sourceforge.net

Provos, Ni. (2004). A Virtual Honeypot Framework. https://www.usenix.org/legacy/event/sec04/tech/full_papers/provos/provos_html/

Serbanescu, A., Obermeier, S., & Yu, D. (2015). A Flexible Architecture for Industrial Control System Honeypots. Proceedings of the 12th International Conference on Security and Cryptography.

Sk4ld (2015). GridPot (Version 2) [Computer software]. Github. https://github.com/sk4ld/GridPot

Yen, K. (2019). BlueKeep: Detecting and Remediating a Critical and Wormable Remote Code Execution Vulnerability. https://www.opswat.com/blog/bluekeep-detecting-and-remediating-a-critical-and-wormable-remote-code-execution-vulnerability

Zorz, Z. (2022). Sandworm attackers tried (and failed) to disrupt Ukraine's power grid. HelpNetSecurity, April 12, 2022.www.helpnetsecurity.com/2022/04/12//sandworm-ukraine/